

IMPLEMENTASI METODE *BACKPROPAGATION* UNTUK PREDIKSI HARGA BATU BARA

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Miracle Fachrunnisa Almas

NIM: 145150200111057



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

IMPLEMENTASI METODE *BACKPROPAGATION*
UNTUK PREDIKSI HARGA BATU BARA

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer


Disusun Oleh :
Miracle Fachrunnisa Almas
NIM: 145150200111057

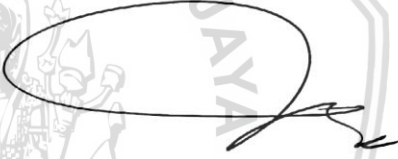
Skripsi ini telah diuji dan dinyatakan lulus pada
30 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I


Dosen Pembimbing II


Budi Darma Setiawan, S.Kom., M.Cs
NIP: 19841015 201404 1 002


Ir. Sutrisno, MT
NIP: 19570325 198701 1 001

Mengetahui
Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

IDENTITAS TIM PENGUJI

Penguji I (Ketua Majelis): Muhammad Tanzil Furqon, S.Kom, M.CompSc

Penguji II : Imam Cholissodin, S.Si, M.Kom



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Juni 2018



Miracle Fachrunnisa Almas

NIM: 145150200111057

KATA PENGANTAR

Puji syukur atas kehadiran Tuhan Yang Maha Esa yang memberikan berkah dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul Implementasi Metode *Backpropagation* Untuk Prediksi Harga Batu Bara dengan baik dan tepat waktu.

Penulis dapat menuntaskan skripsi ini berkat bimbingan dan dukungan dari pihak-pihak terkait yang telah banyak membantu penyelesaian dari skripsi ini. Untuk itu pada kesempatan kali ini, penulis ingin menyampaikan rasa terima kasih kepada:

1. Bapak Budi Darma Setiawan, S.Kom, M.Cs selaku pembimbing I dan Bapak Ir. Sutrisno, M.T selaku pembimbing II yang senantiasa memberikan bimbingan, wejangan dan saran kepada penulis selama penyusunan skripsi.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, P.Hd selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Ir. Heru Nurwasito, M.Kom selaku Wakil Dekan I Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Suprpto, S.T, M.T selaku Wakil Dekan II Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
6. Seluruh jajaran Dosen dan Staf di FILKOM UB yang telah memberikan banyak ilmu dan bantuan kepada penulis selama menempuh perkuliahan.
7. Orang tua penulis yang senantiasa memberi dukungan berupa moril maupun materiil, serta doa dan semangat kepada penulis.
8. Dicky Sidharta yang selalu memberikan semangat, dukungan dan doa selama penulisan skripsi.
9. Ulfani dan teman-teman Kapak GL yang telah memberikan dukungan dan semangat kepada penulis.

Penulis menyadari bahwa skripsi ini masih memiliki banyak kekurangan dan perlu penyempurnaan, oleh karena itu dengan segala kerendahan hati penulis meminta kritik dan saran yang bersifat membangun untuk menyempurnakan skripsi ini. Diharapkan skripsi ini dapat berguna serta bermanfaat bagi seluruh pihak dan bagi penulis sendiri.

Malang, 17 Juni 2018

Penulis

miraclealmas@gmail.com

ABSTRAK

Miracle Fachrunnisa Almas, Implementasi Metode *Backpropagation* Untuk Prediksi Harga Batu Bara

Pembimbing: Budi Darma Setiawan, S.Kom, M.Cs dan Ir. Sutrisno, MT

Batu bara merupakan sumber daya alam yang termasuk dalam salah satu bahan bakar fosil. Indonesia sendiri merupakan salah satu negara dengan produksi dan eksportir batu bara terbesar di dunia. Batu bara telah menjadi komponen penting dalam berjalannya sebuah perusahaan industri berskala besar sebagai bahan bakar industri. Prediksi harga batu bara dibutuhkan karena apabila menunggu harga batu bara yang dirilis oleh pemerintah biasanya akan membutuhkan waktu yang lama. Data harga batu bara bersifat *time series*. Data yang digunakan adalah data harga batu bara dimulai dari bulan Januari tahun 2009 hingga bulan September tahun 2017 dengan merk dagang Gunung Bayan I. Penelitian ini membahas metode *Backpropagation* yang digunakan untuk memprediksi harga batu bara. Pada penelitian ini, dapat diketahui pengaruh perubahan nilai parameter dari *Backpropagation* dalam memprediksi harga batu bara. Output yang dihasilkan oleh sistem berupa hasil prediksi harga batu bara pada bulan selanjutnya. Dari hasil pengujian didapatkan nilai MSE (*Mean Square Error*) terendah sebesar 0,00205284 dengan kombinasi 10 neuron pada *input layer*, 10 neuron pada *hidden layer*, 1 neuron yang dihasilkan sebagai output, *learning rate* sebesar 0,1 dan jumlah iterasi sebesar 500.

Kata kunci: prediksi, batu bara, jaringan syaraf tiruan, *backpropagation*, *time series*

ABSTRACT

Miracle Fachrunnisa Almas, Implementation of Backpropagation Method For Coal Price Prediction

Supervisor: Budi Darma Setiawan, S.Kom, M.Cs and Ir. Sutrisno, MT

Coal is a natural resource that belongs to one of the fossil fuels. Indonesia is one of the countries with the largest quantity of coal production and export in the world. Coal becomes an important component in the running of a large-scale industrial company as an industrial fuel. Predicted coal prices are needed because coal prices released by the government usually takes a long time. Coal price data is in the form of time series. The data used is coal price data starting from January 2009 to September 2017 with trademark of Gunung Bayan I. This research discusses Backpropagation method that is used to predict the coal price. In this research, the effect of change parameter value from Backpropagation in predicting coal price it can be seen. Output generated by the system is in the form of predicted coal price in the next month. The results of the tests are, the lowest MSE (Mean Square Error) value of 0,00205284 with a combination of 10 neurons on the input layer, 10 neurons in the hidden layer, 1 neuron produced as output, learning rate of 0.1 and the number of iterations of 500.

Keywords: prediction, coal, neural network, backpropagation, time series

DAFTAR ISI

PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	v
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR KODE PROGRAM	xiv
DAFTAR LAMPIRAN	viii
BAB 1 PENDAHULUAN.....	Error! Bookmark not defined.
1.2 Latar belakang.....	Error! Bookmark not defined.
1.3 Rumusan masalah	Error! Bookmark not defined.
1.4 Tujuan	Error! Bookmark not defined.
1.5 Manfaat.....	Error! Bookmark not defined.
1.6 Batasan masalah	Error! Bookmark not defined.
1.7 Sistematika pembahasan	Error! Bookmark not defined.
BAB 2 LANDASAN KEPUSTAKAAN	Error! Bookmark not defined.
2.2 Kajian Pustaka	Error! Bookmark not defined.
2.2 Batu Bara.....	Error! Bookmark not defined.
2.2.1 Proses Terbentuknya Batu bara	Error! Bookmark not defined.
2.2.2 Produksi dan Ekspor Batu bara di Indonesia	Error! Bookmark not defined.
2.2.3 Harga Batu bara di Indonesia	Error! Bookmark not defined.
2.3 Prediksi.....	Error! Bookmark not defined.
2.4 Jaringan Syaraf Tiruan (JST)	Error! Bookmark not defined.
2.4.1 Arsitektur Jaringan Syaraf Tiruan	Error! Bookmark not defined.
2.5 <i>Backpropagation</i>	Error! Bookmark not defined.
2.5.1 Algoritma Proses <i>Training</i> pada <i>Backpropagation</i>	Error! Bookmark not defined.

2.5.2 Fungsi Aktivasi	Error! Bookmark not defined.
2.6 Mean Square Error (MSE)	Error! Bookmark not defined.
2.7 Normalisasi dan Denormalisasi	Error! Bookmark not defined.
2.8 Stop Condition	Error! Bookmark not defined.
BAB 3 METODOLOGI	Error! Bookmark not defined.
3.1 Studi Literatur	Error! Bookmark not defined.
3.2 Pengumpulan Data	Error! Bookmark not defined.
3.3 Perancangan	Error! Bookmark not defined.
3.4 Implementasi	Error! Bookmark not defined.
3.5 Pengujian	Error! Bookmark not defined.
3.6 Penarikan Kesimpulan dan Saran	Error! Bookmark not defined.
BAB 4 PERANCANGAN	Error! Bookmark not defined.
4.1 Deskripsi Umum Sistem	Error! Bookmark not defined.
4.2 Diagram Alir Sistem	Error! Bookmark not defined.
4.2.1 Diagram Alir Normalisasi Data	Error! Bookmark not defined.
4.2.2 Diagram Alir Algoritma <i>Backpropagation</i>	Error! Bookmark not defined.
4.2.3 Diagram Alir Proses Denormalisasi Data	Error! Bookmark not defined.
4.2.4 Diagram Alir Perhitungan MSE	Error! Bookmark not defined.
4.3 Perhitungan Manual	Error! Bookmark not defined.
4.3.1 Penginputan Data	Error! Bookmark not defined.
4.3.2 Inisialisasi Bobot Awal dan Bias	Error! Bookmark not defined.
4.3.3 <i>Feedforward</i>	Error! Bookmark not defined.
4.3.4 <i>Backpropagation</i>	Error! Bookmark not defined.
4.3.5 <i>Weight Update</i> (Perubahan Bobot)	Error! Bookmark not defined.
4.3.6 Pengujian	Error! Bookmark not defined.
4.4 Perancangan Antarmuka	Error! Bookmark not defined.
4.5 Perancangan Pengujian	Error! Bookmark not defined.
4.5.1 Pengujian Jumlah Neuron pada <i>Input Layer</i> dan <i>Hidden Layer</i> Terbaik	Error! Bookmark not defined.
4.5.2 Pengujian Nilai <i>Learning Rate</i> (α)	Error! Bookmark not defined.

4.5.3 Pengujian Variasi Jumlah Data <i>Training</i>	Error! Bookmark not defined.
4.5.4 Pengujian Variasi Jumlah Iterasi	Error! Bookmark not defined.
BAB 5 IMPLEMENTASI	Error! Bookmark not defined.
5.1 Implementasi Sistem.....	Error! Bookmark not defined.
5.1.1 Implementasi Proses Normalisasi Data	Error! Bookmark not defined.
5.1.2 Implementasi Proses Inisialisasi Bobot dan Bias	Error! Bookmark not defined.
5.1.3 Implementasi Proses Feedforward .	Error! Bookmark not defined.
5.1.4 Implementasi Proses Backpropagation	Error! Bookmark not defined.
5.1.5 Implementasi Proses Update Bobot dan Bias	Error! Bookmark not defined.
5.1.6 Implementasi Proses <i>Training</i>	Error! Bookmark not defined.
5.1.7 Implementasi Proses Testing	Error! Bookmark not defined.
5.1.8 Implementasi Proses Denormalisasi Data	Error! Bookmark not defined.
5.1.9 Implementasi Evaluasi Sistem Menggunakan MSE.....	Error! Bookmark not defined.
5.2 Implementasi Antarmuka	Error! Bookmark not defined.
BAB 6 PENGUJIAN	Error! Bookmark not defined.
6.1 Pengujian Jumlah Neuron Input dan Hidden Terbaik.	Error! Bookmark not defined.
6.2 Pengujian Learning Rate Terbaik	Error! Bookmark not defined.
6.3 Pengujian Variasi Jumlah Data Training...	Error! Bookmark not defined.
6.4 Pengujian Variasi Jumlah Iterasi	Error! Bookmark not defined.
BAB 7 PENUTUP	70
7.1 Kesimpulan.....	70
7.2 Saran	70
DAFTAR PUSTAKA.....	70

DAFTAR TABEL

Tabel 2.1 Tabel Perbandingan Objek dan Metode ...	Error! Bookmark not defined.
Tabel 2.2 Tabel produksi, ekspor dan konsumsi batu bara di Indonesia.....	Error! Bookmark not defined.
Tabel 2.3 Tabel HBA Indonesia.	Error! Bookmark not defined.
Tabel 3.1 Data Harga Patokan Batu Bara Gunung Bayan I	Error! Bookmark not defined.
Tabel 4.1 Data Asli Harga Batu Bara	Error! Bookmark not defined.
Tabel 4.2 Data Input Harga Batu Bara.....	Error! Bookmark not defined.
Tabel 4.3 Data Input Hasil Normalisasi	Error! Bookmark not defined.
Tabel 4.4 Bobot V_{ij}	Error! Bookmark not defined.
Tabel 4.5 Bobot V_{0j}	Error! Bookmark not defined.
Tabel 4.6 Bobot W_{jk}	Error! Bookmark not defined.
Tabel 4.7 Nilai Hasil Z_{inj}	Error! Bookmark not defined.
Tabel 4.8 Nilai Hasil Z_j	Error! Bookmark not defined.
Tabel 4.9 Nilai Hasil Y_{ink}	Error! Bookmark not defined.
Tabel 4.12 Nilai Hasil ΔW_{jk}	Error! Bookmark not defined.
Tabel 4.13 Nilai Hasil δ_{inj}	Error! Bookmark not defined.
Tabel 4.15 Nilai Hasil ΔV_{ij}	Error! Bookmark not defined.
Tabel 4.16 Nilai Bobot w_{jkbaru}	Error! Bookmark not defined.
Tabel 4.17 Tabel Input Harga Batu Bara (Data <i>Testing</i>)	Error! Bookmark not defined.
Tabel 4.18 Hasil Pengujian	Error! Bookmark not defined.
Tabel 4.19 Pengujian Jumlah Neuron pada <i>Input Layer</i> dan <i>Hidden Layer</i> Terbaik	Error! Bookmark not defined.
Tabel 4.20 Pengujian Jumlah <i>Learning Rate</i>	Error! Bookmark not defined.
Tabel 4.21 Pengujian Variasi Jumlah Data Training ..	Error! Bookmark not defined.
Tabel 4.22 Pengujian Variasi Jumlah Iterasi.....	Error! Bookmark not defined.
Tabel 6.1 Nilai Error Percobaan Pertama.....	Error! Bookmark not defined.
Tabel 6.2 Nilai Error Percobaan Kedua	Error! Bookmark not defined.
Tabel 6.3 Nilai Error Percobaan Ketiga	Error! Bookmark not defined.
Tabel 6.4 Nilai Error Percobaan Keempat.....	Error! Bookmark not defined.

Tabel 6.5 Nilai Error Percobaan Kelima	Error! Bookmark not defined.
Tabel 6.6 Rata-Rata Nilai Error	Error! Bookmark not defined.
Tabel 6.7 Hasil Pengujian Nilai Learning Rate Terbaik	Error! Bookmark not defined.
Tabel 6.8 Hasil Pengujian Data Training.....	Error! Bookmark not defined.
Tabel 6.9 Hasil Pengujian Jumlah Iterasi	Error! Bookmark not defined.



DAFTAR GAMBAR

Gambar 2.1 Arsitektur <i>Single Layer Network</i>	Error! Bookmark not defined.
Gambar 2.2 Arsitektur <i>Multilayer Network</i>	Error! Bookmark not defined.
Gambar 2.3 Arsitektur Competitive Layer Network .	Error! Bookmark not defined.
Gambar 2.4 Arsitektur Backpropagation	Error! Bookmark not defined.
Gambar 3.1 Diagram Alir Metodologi	Error! Bookmark not defined.
Gambar 4.1 Arsitektur Jaringan Syaraf Tiruan <i>Backpropagation</i>	Error! Bookmark not defined.
Gambar 4.2 Diagram Alir Sistem	Error! Bookmark not defined.
Gambar 4.3 Diagram Alir Proses Normalisasi	Error! Bookmark not defined.
Gambar 4.4 Diagram Alir Algoritma <i>Backpropagation</i>	Error! Bookmark not defined.
Gambar 4.5 Diagram Alir Proses <i>Feedforward</i>	Error! Bookmark not defined.
Gambar 4.6 Diagram Alir Perhitungan nilai z	Error! Bookmark not defined.
Gambar 4.7 Diagram Alir Perhitungan nilai y	Error! Bookmark not defined.
Gambar 4.8 Diagram Alir Perhitungan <i>Backpropagation Error</i>	Error! Bookmark not defined.
Gambar 4.9 Diagram Alir Perhitungan nilai δ_k	Error! Bookmark not defined.
Gambar 4.10 Diagram Alir Perhitungan nilai ΔW_{jk} dan ΔW_{ok}	Error! Bookmark not defined.
Gambar 4.11 Diagram Alir Perhitungan nilai δ_{in_j} dan δ_j	Error! Bookmark not defined.
Gambar 4.12 Diagram Alir Perhitungan nilai ΔV_{ij} dan ΔV_{oj}	Error! Bookmark not defined.
Gambar 4.13 Diagram Alir Perhitungan <i>Update Bobot dan Bias</i>	Error! Bookmark not defined.
Gambar 4.14 Diagram Alir Perhitungan <i>Update Bobot dan Bias</i> (w_{jk} dan w_{ok})	Error! Bookmark not defined.
Gambar 4.15 Diagram Alir Perhitungan <i>Update Bobot dan Bias</i> (V_{jk} dan V_{ok})	Error! Bookmark not defined.
Gambar 4.16 Diagram Alir Proses Denormalisasi	Error! Bookmark not defined.
Gambar 4.17 Diagram Alir Perhitungan MSE	Error! Bookmark not defined.
Gambar 4.15 Rancangan Antarmuka Sistem	Error! Bookmark not defined.

Gambar 5.1 Antarmuka Input Data.....**Error! Bookmark not defined.**

Gambar 5.2 Antarmuka *Training***Error! Bookmark not defined.**

Gambar 5.3 Antarmuka Hasil Prediksi**Error! Bookmark not defined.**

Gambar 6.1 Grafik Hasil Pengujian Jumlah *Learning Rate***Error! Bookmark not defined.**

Gambar 6.2 Grafik Hasil Pengujian Jumlah Data *Training***Error! Bookmark not defined.**

Gambar 6.3 Grafik Hasil Pengujian Jumlah Iterasi**Error! Bookmark not defined.**



DAFTAR KODE PROGRAM

Kode Program 5.1 Proses Normalisasi Data.....	Error! Bookmark not defined.
Kode Program 5.2 Proses Inialisasi Bobot dan Bias.....	Error! Bookmark not defined.
Kode Program 5.3 Proses Feedforward	Error! Bookmark not defined.
Kode Program 5.4 Proses Backpropagation.....	Error! Bookmark not defined.
Kode Program 5.5 Proses Update Bobot dan Bias	Error! Bookmark not defined.
Kode Program 5.6 Proses <i>Training</i>	Error! Bookmark not defined.
Kode Program 5.7 Proses <i>Testing</i>	Error! Bookmark not defined.
Kode Program 5.8 Proses Denormalisasi Data.....	Error! Bookmark not defined.
Kode Program 5 .9 Evaluasi Sistem Menggunakan MSE.....	Error! Bookmark not defined.



DAFTAR LAMPIRAN

LAMPIRAN A DATA HARGA BATU BARA	72
A.1 Data Harga Batu Bara Asli.....	72
LAMPIRAN B WAWANCARA DENGAN NARASUMBER	77
B.1 Konsumsi Batu Bara di Indonesia	77





BAB 1 PENDAHULUAN

1.2 Latar belakang

Batu bara merupakan sumber daya alam yang termasuk dalam salah satu bahan bakar fosil. Proses pembentukan batubara berasal dari sisa-sisa tanaman dan membutuhkan waktu berjuta-juta tahun. Batu bara memiliki sifat-sifat kimia dan fisika yang kompleks dan tersusun dari unsur oksigen, hidrogen serta karbon. Batu bara adalah salah satu dari sekian banyak bahan bakar yang bersifat hidro-karbon padat yang terbentuk dari berbagai macam tumbuhan yang terkena pengaruh dari tekanan temperatur pada lingkungan terbebas dari oksigen (Priyono, et al.,1992).

Sejak tahun 2005, Indonesia termasuk ke dalam salah satu negara dengan penghasil batu bara dan eksportir terbesar di dunia. Tiga daerah di Indonesia yang memiliki cadangan batubara terbesar antara lain yaitu Kalimantan Timur, Kalimantan Selatan dan Sumatera Selatan. Pada industri batu bara yang ada di Indonesia terbagi menjadi dua bagian yaitu industri produsen dengan skala besar dan industri produsen dengan skala kecil yang memiliki tambang batu bara terbagi di daerah Sumatera dan Kalimantan. Berkaitan dengan cadangan batu bara saat ini, BP *Statistical Review of World Energy* Indonesia menyatakan bahwa Indonesia memiliki 2,2 persen cadangan batu bara serta menempati tempat ke-9 (BP, 2017).

Dalam kehidupan sehari-hari, tanpa kita sadari batu bara memiliki banyak manfaat penting. Masyarakat umumnya tidak memanfaatkan batu bara secara langsung, namun biasanya perusahaan industri berskala besar yang menjadikan batu bara sebagai bahan bakar utama. Manfaat batu bara diantaranya adalah sebagai bahan bakar pabrik kertas, bahan bakar dalam industri produksi semen, penghasil fenol (salah satu produk pada bahan bakar jenis minyak) dan lain-lain.

Dikarenakan batu bara telah menjadi komponen penting dalam berjalannya sebuah perusahaan industri berskala besar, dapat dipastikan perusahaan akan secara berkala membeli batu bara dari produsen batu bara yang telah mereka percayai akan kualitas batu bara yang dihasilkan. Dibutuhkan suatu perkiraan harga dikarenakan harga batu bara yang selalu berubah-ubah setiap waktunya. Hal ini dilakukan karena apabila menunggu harga batu bara yang dirilis oleh pemerintah biasanya akan membutuhkan waktu yang lama.

Terdapat macam-macam metode yang dapat kita gunakan untuk implementasi suatu sistem prediksi dengan hasil yang baik dan akurat. Metode *Backpropagation* merupakan salah satu metode yang biasa digunakan untuk sistem prediksi. Metode *Backpropagation* merupakan suatu algoritma pembelajaran (*Supervised Learning*) yang digunakan untuk memperkecil nilai tingkat error yaitu dengan cara melakukan penyesuaian bobot berdasar perbedaan dari target dan output ingin dicapai. Metode

ini biasa digunakan dalam suatu sistem prediksi. Menurut Haykin (1999) *backpropagation* merupakan suatu teknik atau ilmu yang berguna untuk meminimalisir *gradient* pada sebuah dimensi bobot dengan proses *training* yang dilakukan secara terulang pada jaringan syaraf tiruan dengan banyak *layer* (lapisan).

Terdapat beberapa penelitian mengenai metode *Backpropagation* yang penulis gunakan sebagai referensi pada penelitian ini. Penelitian yang pertama adalah penelitian dari Suryaningrum, Ratnawati dan Setiawan (2017) yang melakukan penelitian tentang prediksi waktu panen tebu menggunakan gabungan metode *Backpropagation* dan Algoritma Genetika. Penelitian ini bertujuan untuk mengetahui waktu yang tepat digunakan dalam melakukan panen dikarenakan sebelum tebu digiling, diperlukan analisis tingkat kematangan dari tumbuhan tebu. Algoritma genetika pada penelitian ini dipergunakan dalam pengoptimalisasi pemilihan bobot serta bias. Sedangkan metode *backpropagation* digunakan untuk menghitung nilai *Mean Square Error* (MSE) yang digunakan dalam perhitungan fitness dan juga pada proses prediksi data uji. Nilai ini nantinya akan digunakan pula untuk mencari nilai *fitness* dan untuk proses prediksi dari data uji. Dari penelitian ini diperoleh hasil pengujian terbaik yaitu nilai AFER sebesar 0,0205%.

Selanjutnya adalah penelitian yang dilakukan oleh Wanto dan Windarto (2017) tentang menganalisis prediksi IHK (Indeks Harga Konsumen) menggunakan metode *Backpropagation*. Penghitungan IHK bertujuan untuk mengetahui perubahan-perubahan yang terjadi pada harga dari sekelompok jasa dan barang yang oleh masyarakat secara umum dikonsumsi.

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Susanti (2014) yaitu melakukan prediksi harga ayam menggunakan metode *Backpropagation*. Pada penelitian ini, data yang dipergunakan adalah data dengan jenis *time series* atau data deret waktu dari bulan Agustus 2010 hingga bulan Mei 2013. Peramalan harga ayam berperan penting pada industri peternakan unggas karena memiliki manfaat yang dihasilkan dapat memaksimalkan keuntungan serta meminimalisir resiko. Diperoleh hasil penelitian berupa arsitektur jaringan 4-10-1, yakni pada lapisan *input* dengan 4 neuron, pada lapisan *hidden* dengan 10 neuron dan pada lapisan output dengan 1 neuron. Hasil penelitian tersebut merupakan hasil penelitian yang optimal dengan menggunakan metode *Backpropagation*.

Berdasarkan permasalahan yang telah diuraikan dan dengan adanya referensi-referensi di atas yang dapat dipergunakan sebagai perbandingan, maka akan diusulkan penelitian yaitu mengenai sistem yang dapat memprediksi harga batu bara menggunakan metode *Backpropagation*. Data yang digunakan adalah data harga dari bulan Januari 2009 hingga bulan September 2017 dengan merk dagang Gunung Bayan I. Diharapkan *output* dari sistem yang berupa prediksi harga batu bara selanjutnya

mampu menjawab permasalahan konsumen mengenai perkiraan harga batu bara yang dibutuhkan.

1.3 Rumusan masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat diperoleh rumusan masalah sebagai berikut:

1. Bagaimana pengaruh perubahan nilai parameter *Backpropagation* terhadap hasil akurasi dari sistem dalam memprediksi harga batu bara menggunakan metode *Backpropagation*?
2. Bagaimana hasil nilai *error* dari pengujian yang didapatkan dari sistem prediksi harga batu bara dengan menggunakan perhitungan *Mean Square Error* (MSE)?

1.4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Bertujuan untuk mengetahui pengaruh dari perubahan nilai parameter *Backpropagation* terhadap hasil akurasi dari sistem dalam memprediksi harga batu bara menggunakan metode *Backpropagation*
2. Melakukan pengujian dan mengetahui hasil nilai *error* yang dihasilkan oleh sistem melalui perhitungan *Mean Square Error* (MSE).

1.5 Manfaat

Adapun manfaat yang dapat diperoleh dari penelitian ini adalah:

1. Menerapkan ilmu dan kemampuan yang telah diperoleh selama di bangku perkuliahan.
2. Mengembangkan kemampuan dalam merancang sebuah sistem prediksi guna membantu masyarakat luas.
3. Membantu konsumen yang membutuhkan perkiraan harga batu bara ini untuk memperoleh perkiraan harga yang sesuai.

1.6 Batasan masalah

Dalam penelitian ini masih terdapat beberapa batasan diantaranya:

1. Data yang digunakan adalah data harga batu bara dengan merk dagang Gunung Bayan I dimulai pada bulan Januari 2009 hingga September 2017.
2. Dalam melakukan implementasi, sistem menggunakan metode *Backpropagation*.

3. Pengujian nilai error yang dilakukan menggunakan perhitungan *Mean Square Error* (MSE).
4. Hanya terdapat 1 fitur yang digunakan yaitu harga batu bara secara bulanan dimulai dari bulan Januari 2009 hingga September 2017.

1.7 Sistematika pembahasan

Berikut merupakan sistematika pembahasan yang digunakan pada penelitian ini, termasuk mencakup pada setiap babnya:

BAB I PENDAHULUAN

Pada bab I yaitu Pendahuluan akan berisi mengenai sub bab diantaranya membahas mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi tentang ulasan mengenai teori-teori dari sumber pustaka dan referensi yang berkaitan dengan teori-teori pembuatan sistem “Implementasi Metode *Backpropagation* Untuk Prediksi Harga Batu Bara” ini seperti penjelasan mengenai batu bara, manfaat batu bara, prediksi serta metode yang digunakan yaitu metode *Backpropagation*.

BAB III METODOLOGI

Bab ini berisi tentang urutan dalam langkah kerja yang dilakukan dalam penulisan skripsi. Sub bab dari bab ini terdiri dari studi literatur, pengumpulan data, analisis kebutuhan, perancangan, implementasi, pengujian dan analisis, dan pada tahap akhir dari bab ini berisi kesimpulan.

BAB IV PERANCANGAN

Bab ini membahas mengenai perancangan sistem yang terdiri dari deskripsi umum sistem, diagram alir sistem, perancangan manualisasi, perancangan antarmuka, perancangan uji coba serta perancangan evaluasi sistem.

BAB V IMPLEMENTASI

Bab 5 yaitu bab Implementasi, ini berisi mengenai implementasi dari sistem “Implementasi Metode *Backpropagation* Untuk Prediksi Harga Batu Bara”. Implementasi dilakukan berdasarkan dari perancangan dari bab sebelumnya.

BAB VI PENGUJIAN DAN ANALISIS

Pada Bab 6 yaitu pengujian dan analisis, ini berisi mengenai perincian bagaimana pengujian yang telah dilakukan serta memberikan analisis yang tepat sesuai dengan hasil yang diperoleh.

BAB VII PENUTUP

Bab ini berisi mengenai kesimpulan yang telah didapatkan dari hasil implementasi serta pengujian sistem pada yang telah dilakukan, serta terdapat saran agar dapat dipergunakan untuk membangun dan mengembangkan penelitian selanjutnya sehingga lebih baik dari penelitian ini.



BAB 2 LANDASAN KEPUSTAKAAN

Dalam bab ini, akan dijelaskan mengenai dasar teori dan kajian pustaka yang berasal dari jurnal, buku dan hasil penelitian sebelumnya yang digunakan dan berkaitan guna mendukung penulisan skripsi ini.

1.1 Kajian Pustaka

Terdapat banyak penelitian sebelumnya mengenai prediksi menggunakan metode *backpropagation* yang diantaranya adalah penelitian oleh Suryaningrum, Ratnawati dan Setiawan (2017) tentang prediksi dari waktu panen tebu. Melakukan suatu analisis terhadap tingkat kematangan buah tebu dibutuhkan sebelum tebu digiling. Cara yang paling mudah untuk melihat tingkat kematangan tebu adalah dengan melakukan proses prediksi pada waktu pelaksanaan panen tebu. Digunakan 2 metode dalam penelitian ini yaitu Algoritma Genetika dan metode *Backpropagation*. Dimana dalam melakukan proses awal bobot dan bias awal menggunakan algoritma genetika dan untuk proses prediksinya menggunakan metode *Backpropagation*. Pada penelitian ini terdapat 2 metode, gunanya adalah untuk mengoptimasi nilai bobot awal dan bias awal dengan memilih nilai fitness yang paling kecil untuk proses prediksi dari waktu panen tebu. Nilai AFER sebesar 0,0205% didapatkan sebagai hasil pengujian terbaik dari penelitian ini. Hasil pengujian mendekati 0%, maka dapat dinyatakan bahwa sistem baik dalam melakukan hasil prediksi.

Penelitian selanjutnya adalah penelitian dari Wanto dan Windarto (2017) mengenai prediksi IHK (Indeks Harga Konsumen) yang berdasarkan pada kelompok kesehatan dengan menggunakan metode *Backpropagation*. IHK merupakan salah satu indikator penting dalam bidang ekonomi yang dapat memberikan dan menentukan perkembangan daripada harga barang maupun jasa yang di bayar oleh para konsumen di suatu wilayah. Perubahan harga dari sekumpulan tetap barang atau jasa yang umumnya dikonsumsi oleh sebagian besar dari masyarakat pada suatu wilayah dapat diketahui dengan penghitungan IHK (Sinaga, et al.,2014). Data Indeks Harga Konsumen berdasarkan kelompok Kesehatan yang digunakan pada penelitian ini adalah data tahun 2014 hingga 2016 pada bulan Januari-Desember yang berasal dari Badan Pusat Statistik (BPS) Kota Medan. Digunakan 8 model arsitektur menggunakan metode *Backpropagation* pada penelitian ini. Model arsitektur *backpropagation* terbaik yang dihasilkan menggunakan model 12-70-1 yang menghasilkan tingkat akurasi sebesar 92%, MSE 0,3659742 dengan tingkat error yang digunakan 0,001 – 0,05.

Susanti (2014) melakukan penelitian untuk meramalkan harga ayam dengan menerapkan model *backpropagation NN (neural networks)*. Prediksi harga secara

tepat pada sektor unggas mampu memberikan peningkatan efisiensi, mengoptimalkan sumber daya, dan meningkatkan pendapatan. Data masukan yang digunakan adalah data harga ayam ke-(t) 7, 14, 21 sedangkan untuk target yang akan diperoleh adalah data harga ayam ke-(t+7). Hasil dari penelitian ini menunjukkan bahwa, arsitektur jaringan 4-10-1, yakni 1 lapisan pada unit input dengan 4 neuron, 1 lapisan pada unit hidden dengan 10 neuron dan 1 lapisan unit output dengan 1 neuron merupakan jaringan aksitetur yang cukup optimal. Fungsi aktivasi tansig dan fungsi pelatihan trainrp (*Resilient Bacpropagation*) dengan hasil toleransi error sebanyak 0,001, learning rate sebesar 0,05 serta 5000 pada nilai maksimum epoch yang digunakan. Nilai koefisien korelasi untuk data pengujian adalah senilai 0,8696, nilai pada MSE yang dihasilkan sebesar 0,0113 dan nilai koefisien korelasi pada data pelatihan yaitu sebanyak 0,961661.

Penelitian yang ke-4 adalah penelitian dari Triyono, Santoso dan Pranowo (2016) yaitu tentang peramalan harga saham (IHSG) menggunakan metode *backpropagation*. Data catatan yang digunakan yaitu harga pembukaan, terendah, tertinggi, dan penutupan, data-data tersebut merupakan data IHSG sendiri. Data yang digunakan berasal dari data harian dari tahun 2010-2015. Selain itu ditambahkan pula data hasil catatan berupa harga emas dan harga minyak pada hari yang berkaitan. Data-data tersebut merupakan data yang mampu memberikan dampak dari pergerakan harga saham. Algoritma *Backpropagation* adalah algoritma yang merubah nilai bobot dengan menggunakan nilai *error* dari *output* bobot dengan arah mundur. Hasil *Error* akan diperoleh saat proses *forward* terlebih dulu diproses. Tingkat keakuratan yang diperoleh sebanyak 99,98% dengan hasil nilai MSE yaitu 0,9915. Dapat diketahui dari hasil yang diperoleh pada penelitian ini bahwa tingkat keakuratan dalam peramalan harga IHSG menggunakan metode *backpropagation* dapat meningkat dikarenakan penambahan dari variabel harga emas dan harga minyak.

Kemudian penelitian yang dilakukan oleh Lahmiri (2014) yang menggunakan komponen wavelet frekuensi rendah (low frequency) dan frekuensi tinggi (high frequency) sebagai fitur untuk memprediksi harga saham menggunakan *backpropagation neural networks*. Penelitian ini merupakan penelitian bertipe peramalan (forecasting) dimana pada penelitian ini mengintegrasikan Discrete Wavelet Transform (DWT) dengan metode *Backpropagation Neural Networks* (BPNN) untuk melakukan prediksi seri waktu keuangan. Komponen frekuensi tinggi (high frequency) dapat menangkap diskontinuitas, kerusakan dan keganjilan pada data aslinya sedangkan komponen frekuensi rendah (low frequency) memiliki karakteristik memperkasar struktur dari data yang gunanya untuk mengidentifikasi data asli pada waktu jangka panjang. Kedua komponen ini saling melengkapi satu sama lain. Terdapat 7 buah dataset yang diaplikasikan dengan model ini. Untuk semua ukuran

akurasi dataset menunjukkan bahwa model yang disajikan melebihi model konvensional yang hanya menggunakan komponen frekuensi rendah.

Berikut merupakan tabel perbandingan pada objek serta metode pada penelitian sebelumnya, ditunjukkan pada Tabel 2.1.

Tabel 2.1 Tabel Perbandingan Objek dan Metode

No	Judul	Objek	Metode
1	Prediksi Waktu Panen Tebu Menggunakan Gabungan Metode <i>Backpropagation</i> dan Algoritma Genetika (Suryaningrum, et al., 2017).	Indeks Harga Konsumen berdasarkan kelompok kesehatan.	<i>Gabungan Metode Backpropagation dengan Algoritma Genetika.</i>
2	Analisis Prediksi Indeks Harga Konsumen Berdasarkan Kelompok Kesehatan dengan menggunakan Metode <i>Backpropagation</i> (Wanto, et al., 2017).	Indeks Harga Konsumen dari BPS Medan. Data ini berdasar pada kelompok kesehatan dimulai 2014 - 2016.	<i>Backpropagation</i>
3	Penerapan Model <i>Neural Network Backpropagation</i> untuk Prediksi Harga Ayam (Susanti, 2014).	Harga ayam bersifat data time series dari bulan Agustus 2010 – bulan Mei 2013. Terdapat 1015 data.	<i>Backpropagation</i>
4	Penerapan Metode Jaringan Syaraf Tiruan <i>Backpropagation</i> Untuk Meramalkan Harga Saham (IHSG) (Triyono, et al., 2016).	Harga saham pada perusahaan PT. Unilever, PT. ASII Indonesia, PT. Bank BCA, PT. PGAS, PT. Telkom.	<i>Backpropagation</i>
5	<i>Wavelet Low and High-Frequency Components as Features for Predicting Stock Prices with Backpropagation Neural Networks</i> (Lahmiri, 2014).	Prediksi harga saham.	<i>Backpropagation.</i>

2.2 Batu Bara

Batu bara termasuk dalam bahan bakar fosil dan merupakan sumber energi yang penting dalam proses pembangkitan listrik. Akan tetapi batu bara memiliki tingkat polusi tinggi dikarenakan tingginya tingkat kandungan karbon sehingga hal itu memberikan karakter negatif pada batu bara. Gas alam merupakan sumber energi selain batu bara yang memiliki tingkat polusi lebih rendah, namun harga dari gas alam lebih sering mengalami fluktuasi di pasar internasional sehingga semakin banyak perusahaan yang beralih menggunakan batu bara sebagai sumber energi.

Batu bara menurut Muchijidin (2006) adalah satuan batuan sedimen organik berasal dari hasil sisa penguraian bermacam tumbuhan yang merupakan campuran heterogen antar zat organik yang bergabung dibawah beban strata yang telah menghimpitnya. Digunakan dua metode analisis dalam pemisahan batu bara yaitu ekstraksi pelarut dan destilasi destruktif. Dari dua metode tersebut ditunjukkan bahwa batu bara hanya mengandung konsentrasi hidrogen, karbon, nitrogen, oksigen serta sulfur.

1.1.1 Proses Terbentuknya Batu bara

Batu bara terbentuk dari tanaman-tanaman yang telah tertanam di dalam tanah pada waktu yang sangat Panjang, terjaga pada tekanan yang sangat tinggi dan temperatur tertentu. Menurut Diessel (1986) terdapat 2 tahap dalam proses pembentukan batu bara, yaitu:

1. Tahap Biokimia (*Biochemical Stage*)
Tahap pertama dari proses pembentukan batu bara yaitu dengan terjadinya pembusukan sisa tumbuhan-tumbuhan kemudian dilanjutkan dengan proses penggambutan (*peatification*) yang dilakukan oleh mikroba dan bakteri. Proses ini mampu mengakibatkan pelepasan hidrokarbon sehingga mengakibatkan penambahan kandungan karbon pada batu bara.
2. Tahap Fisika-Kimia (*Physico-Chemical Stage*)
Pada tahap ini, gambut yang telah terjadi akibat dari proses sebelumnya akan berubah menjadi berbagai tingkatan dan jenis batu bara. Hal ini karena disebabkan perbedaan waktu geologi, tekanan dan suhu.

Selanjutnya ada 2 teori yang menerangkan pembentukan batu bara Sukandarrumidi (2008) yaitu:

1. Teori In-situ

Teori in-situ menjelaskan bahwa biasanya pembentukan batu bara terjadi di hutan basah yang berawa-rawa, sehingga ketika tumbuh-tumbuhan dan pepohonan mati ataupun roboh akan tenggelam di rawa yang ada disekitar tumbuhan tersebut. Sisa daripada pepohonan dan tumbuhan itu tidak membusuk dengan sempurna. Hingga

berakhir dengan menjadi fosil tumbuhan, sehingga terbentuk berbagai macam sedimen organik. Proses ini menghasilkan batu bara dengan kualitas tinggi / baik.

2. Teori Drift

Pada teori drift dijelaskan bahwa batu bara pada teori terbentuk melalui pepohonan dan tumbuh-tumbuhan yang berasal dari hutan, dimana hutan tersebut bukan merupakan tempat dari batu bara tersebut mengalami pembentukan. Batu bara yang terbentuk melalui teori drift ini memiliki ciri lapisan batu bara tipis, biasanya terbentuk di delta serta banyak zat pengotor (kandungan abunya tinggi), dan banyak lapisannya (*multiple seam*). Kualitas batu bara yang dihasilkan melalui teori ini kurang baik karena bercampur dengan zat pengotor.

1.1.2 Produksi dan Ekspor Batu Bara di Indonesia

Indonesia merupakan negara dengan ekspor batu bara terbesar di dunia. BUMN (2017) mengatakan bahwa sejak tahun 2005, Indonesia mampu menjadi eksportir batu bara thermal setelah produksi batu bara dalam negeri mampu melampaui negara Australia. Porsi signifikan dari batu bara thermal yang diekspor terdiri dari jenis kualitas batu bara dengan tingkat menengah (berkisar 5100-6100 cal/gram) dan jenis kualitas rendah (di bawah 5100 cal/gram). Permintaan ekspor terbanyak sebagian dari negara India serta Cina. Diperkirakan oleh Kementerian Energi dan Sumber Daya Mineral Indonesia bahwa cadangan batu bara habis dalam waktu kurang lebih 82 tahun kedepan jika peningkatan produksi terus menerus dilakukan. Hingga saat ini, permintaan ekspor batu bara ke Cina mengalami pertumbuhan 17 persen dan mencapai US\$20 miliar (BUMN, 2017). Dengan pencapaian tersebut, Cina menjadi negara yang memperoleh ekspor batu bara terbesar dari Indonesia.

Sejak awal tahun 1990 an Indonesia mengalami peningkatan ekspor batu bara. Hal ini dikarenakan sektor pertambangan batu bara semakin gencar dilakukan. Ekspor batu bara di Indonesia berkisar antara 71-80 persen dari total keseluruhan pada produksi batu bara dalam negeri, sedangkan sisanya adalah untuk dijual di pasar domestik. Cadangan batu bara di Indonesia yang melimpah memiliki kualitas menengah. Jenis batu bara yang terdapat di Indonesia dijual dengan harga kompetitif di pasar dunia, serta didukung dengan minimnya upah tenaga kerja di Indonesia sehingga berakibat semakin banyak perusahaan serta para pengusaha untuk berlomba-lomba membuka usaha pertambangan batu bara sejak tahun 2000-an.

Berikut Tabel 2.2 menunjukkan produksi, konsumsi batu bara dan ekspor batu bara di Indonesia dari tahun 2007 hingga 2016.

Tabel 2.2 Tabel produksi, ekspor dan konsumsi batu bara di Indonesia.

	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
Produksi (dalam juta ton)	217	240	254	275	353	412	474	458	461	419
Ekspor (dalam juta ton)	163	191	198	210	287	345	402	382	366	333
Domestik (dalam juta ton)	61	49	56	65	66	67	72	76	87	86

Sumber: Indonesian Coal Mining Association (APBI) & Ministry of Energy and Mineral Resources BUMN (2017).

1.1.3 Harga Batu bara di Indonesia

Skema harga batu bara ditetapkan berdasarkan biaya produksi ditambah dengan margin (keuntungan). Keuntungannya pun harus wajar, telah ditentukan oleh pemerintah yakni sebesar 5%-25%. Kebijakan penentuan harga batu bara di tentukan sesuai dengan peraturan Menteri ESDM No 07 tahun 2017. Peraturan Menteri ini merupakan aturan pelaksanaan dari:

1. Pengganti Peraturan Menteri No. 17 Tahun 2010.
2. Undang-Undang No.4 Tahun 2009 tentang Pertambangan Mineral dan Batu bara.
3. Peraturan Pemerintah No. 23 Tahun 2010 tentang Pelaksanaan Kegiatan Pertambangan Mineral dan Batu bara.

Harga patokan batuan ditentukan oleh pemerintah terdiri atas:

1. HPB (Harga Patokan Batu Bara) untuk *steam (coal)*
2. HPB (Harga Patokan Batu Bara) untuk *coking (metallurgical) coal*

HPB (Harga Patokan Batu bara) ditetapkan oleh Dirjen atas persetujuan dari Menteri. HPB adalah harga di titik serah (*at sale point*) FOB (terminology FOB bisa pada titik dari FOB *Vaessel* maupun pada titik FOB *Barge*). Index-index dari harga yang akan digunakan dalam penentuan HBA (Harga Batu Bara Acuan) adalah pada titik FOB Vessel, sehingga terdapat pilihan 3 sebagai berikut:

1. Karena Indeks Harga yang digunakan untuk penentuan HBA berada pada point FOB Vessel, maka HPB sebagai turunan HBA yang berarti juga HPB pada point FOB Vessel
2. HPB bulanan yang diterbitkan untuk merek terdaftar di minerba akan disesuaikan per jetty/perusahaan berdasarkan pada sertifikat jarak (seluruh perusahaan diwajibkan menyampaikan sertifikat jarak).

Berikut akan ditunjukkan tabel HBA (Harga Acuan Batu Bara) di Indonesia pada tabel 2.3.

Tabel 2.3 Tabel HBA Indonesia.

Bulan	2012	2013	2014	2015	2016	2017
Januari	109.29	87.55	81.90	63.84	53.20	86.23
Februari	111.58	88.35	80.44	62.92	53.20	83.32
Maret	112.87	90.09	77.01	67.76	50.92	81.90
April	105.61	88.56	74.81	64.48	51.62	82.51
Mei	102.12	85.33	73.60	61.08	52.32	83.81
Juni	96.65	84.87	73.64	59.59	51.20	75.46
Juli	87.56	81.69	72.45	59.16	51.87	-
Agustus	84.65	76.70	70.29	59.14	53.00	-
September	98.21	76.89	76.89	58.21	58.37	-
Oktober	86.04	76.61	67.26	57.39	69.07	-
November	81.44	78.13	65.70	54.43	84.86	-
Desember	81.75	80.31	69.23	53.51	101.69	-

Dalam USD/ton

Sumber: Ministry of Energy and Mineral Resources (Permen ESDM, 2017).

2.3 Prediksi

Menurut Heizer dan Render (2001) peramalan atau dalam Bahasa Inggris *forecasting* adalah seni dan keilmuan memprediksi kejadian di masa depan dengan mengambil data historis kemudian dengan menggunakan beberapa bentuk model matematis memproyeksikannya ke masa depan. Sedangkan berdasarkan pendapat Aritonang (2002) peramalan adalah kegiatan penerapan dari model yang telah dikembangkan saat ini pada waktu yang akan datang. Prediksi bertujuan untuk mencari kemungkinan jawaban semirip mungkin yang akan terjadi selanjutnya bukan digunakan untuk mencari jawaban pasti mengenai suatu kejadian yang akan datang (Chandra, Sovia & Permana, 2015).

Terdapat dua macam Teknik prediksi yaitu teknik prediksi kualitatif dan teknik prediksi kuantitatif (Chandra, Sovia & Permana, 2015).

1. Prediksi Kualitatif

Metode ini bersifat *intuitive*, *subjective* dan *judgemental* dikarenakan pada metode prediksi kualitatif ditentukan berdasarkan pengalaman dan pendapat dari penyusunnya. Metode prediksi kualitatif digunakan jika data pada masa

lalu memiliki variabel keterkaitan yang kurang dapat dipercaya dan dipertanggungjawabkan atau bahkan tidak ada sama sekali.

2. Prediksi Kuantitatif

Prediksi kuantitatif dapat digunakan apabila memenuhi 3 hal yaitu, dapat diartikan bahwa pola/kejadian yang telah berlalu akan mengalami kelanjutan pada waktu masa depan, adanya informasi dan berita mengenai keadaan dan kejadian yang lain, berita dan informasi tersebut dapat diproyeksikan dalam bentuk data.

Heizer dan Render (1996) menyatakan bahwa jangka waktu peramalan/ prediksi dapat dikelompokkan menjadi 3 kategori bagian yaitu:

- a. Peramalan jangka pendek
Peramalan yang digunakan pada waktu berjangka kurang dari 3 bulan.
- b. Peramalan Jangka Menengah
Peramalan yang digunakan untuk waktu berjangka antara 3 bulan - 3 tahun.
- c. Peramalan Jangka Panjang
Peramalan yang digunakan untuk waktu dalam jangka lebih dari 3 tahun.

2.4 Jaringan Syaraf Tiruan (JST)

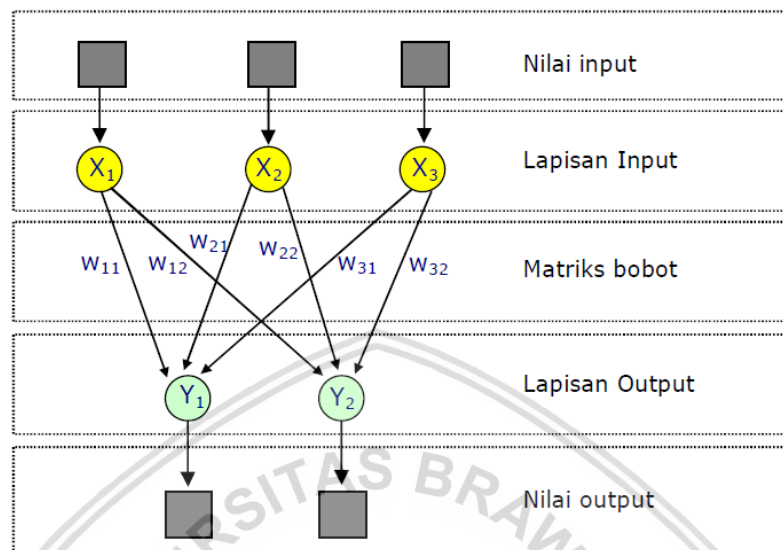
Jaringan syaraf tiruan atau dapat disingkat JST merupakan suatu metode atau ilmu kecerdasan buatan (*Artificial Intelligent*) yang memiliki konsep untuk menyerupai sistem dari jaringan syaraf pada bagian tubuh manusia. Menurut Fausset (1994) jaringan syaraf tiruan adalah sebuah sistem pengolah dan pemberi informasi yang menggunakan konsep serupa dengan jaringan pada syaraf biologis manusia.

Jaringan syaraf tiruan menyelesaikan suatu masalah melalui proses pembelajaran dari contoh pelatihan yang diberikan. Selama waktu proses belajar tersebut, input dan output disajikan dengan cara bersamaan. JST dapat digambarkan sebagai sebuah simulasi (tiruan) dari kumpulan *prototype* saraf biologis. Jadi ilmu jaringan syaraf tiruan adalah ilmu yang termotivasi oleh kesamaan dengan sistem biologis yang berhasil bekerja, yang apabila diperbandingkan dengan sistem secara keseluruhan terdiri dari sel-sel saraf yang sangat sederhana namun banyak yang bekerja secara besar-besaran dan secara paralel serta memiliki kemampuan untuk belajar atau disebut dengan pola pelatihan (Kriesel, 2007).

2.4.1 Arsitektur Jaringan Syaraf Tiruan

Terdapat beberapa jenis arsitektur pada metode Jaringan Syaraf Tiruan. Menurut Haykin (1999), terdapat 3 macam arsitektur pada jaringan syaraf tiruan secara umum yakni:

1. Jaringan syaraf dengan lapisan tunggal (*Single Layer Networks*)

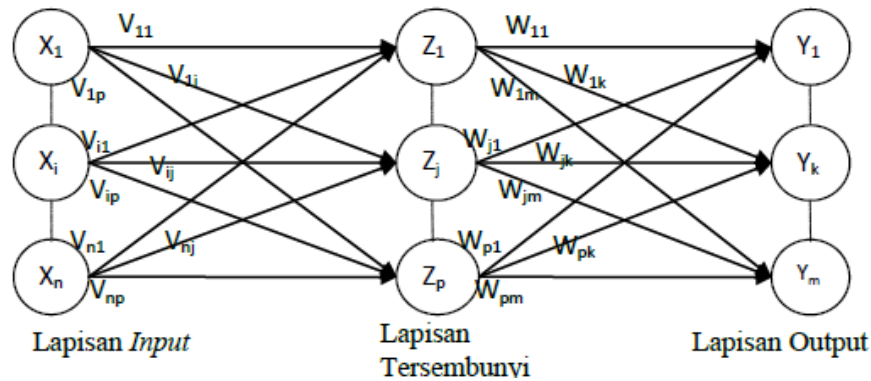


Gambar 2.1 Arsitektur *Single Layer Network*

Pada arsitektur jenis ini, neuron-neuron dikumpulkan dalam layer-layer. Jaringan ini adalah bertipe *feed forward*. Dalam jaringan syaraf tiruan dengan lapisan tunggal terdapat jaringan dengan bentuk paling sederhana, yang terdiri dari sebuah *input layer* dari pusat node pada bagian informasi ditampilkan menuju ke *output layer* yang berasal dari neuron akan tetapi tidak dapat dilakukan, jika dilakukan sebaliknya. Pada *input layer* yang berasal dari pusat node tidak dikalkulasikan karena tidak terdapat suatu proses perhitungan yang dapat dikerjakan.

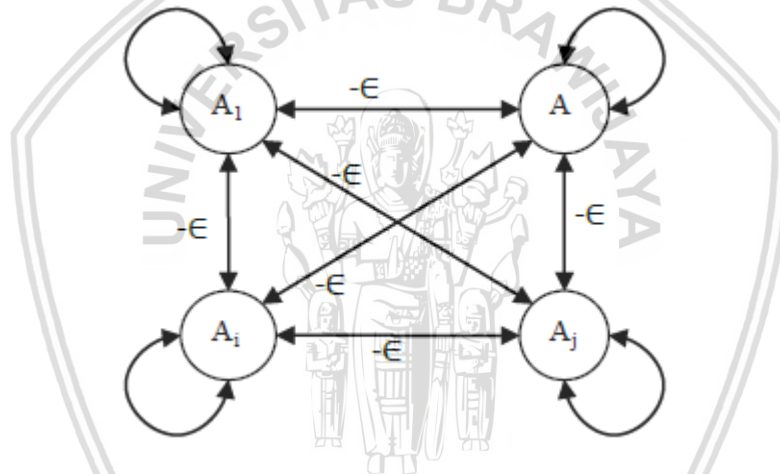
2. Jaringan syaraf dengan banyak lapisan (*Multilayer Networks*)

Multilayer Networks memiliki *hidden layer* atau jaringan tersembunyi. *Multilayer Network* mampu memecahkan masalah yang tidak dapat dipecahkan oleh *Single Layer Networks*, namun arsitektur ini memiliki proses pelatihan yang lebih rumit. Gambar 2.2 dibawah ini akan menunjukkan arsitektur dari *Multilayer Networks*.



Gambar 2.2 Arsitektur *Multilayer Network*

3. Jaringan syaraf dengan lapisan kompetitif (*Competitive Layer Networks*)



Gambar 2.3 Arsitektur *Competitive Layer Network*

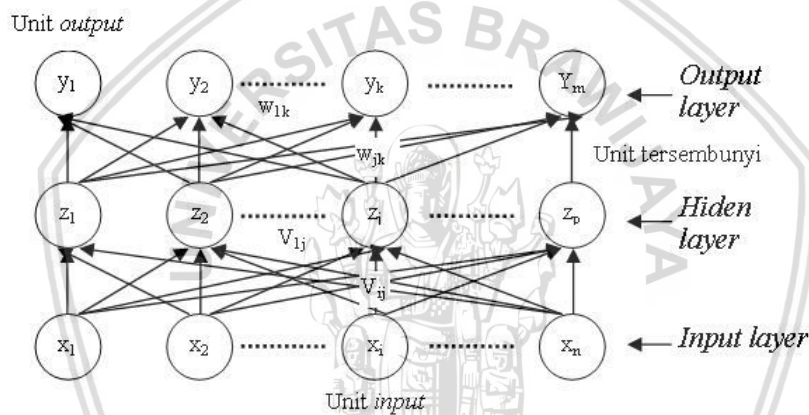
Koneksi yang terjadi pada banyak neuron dalam jaringan lapisan kompetitif tidak dapat digambarkan seperti pada arsitektur jaringan lainnya. Dalam jaringan syaraf ini terdapat sekelompok neuron melakukan persaingan guna memperoleh hak untuk berubah menjadi aktif. Prinsip demikian dapat kita sebut sebagai prinsip *winner takes all* (pemenang dapat mengambil keseluruhannya).

2.5 Backpropagation

Backpropagation adalah algoritma dimana biasanya digunakan oleh perceptron dengan *multi layer* yang bertujuan menjadikan bobot-bobot yang memiliki hubungan dengan neuron yang terdapat pada lapisan *hidden layer*, *backpropagation* merupakan algoritma supervised learning yang terawasi (Tanjung, 2015). *Hidden layer* berperan sebagai detektor fitur, artinya hidden layer berperan sangat penting dalam operasi

perceptron multilayer (Haykin, 2009). *Error* dihitung berdasarkan pada rata-rata kuadrat dari kesalahan atau biasa disebut MSE (*Mean Square Error*).

Haykin (1999) menyatakan bahwa *backpropagation* adalah suatu metode atau teknik untuk meminimalisasi tingkat kemiringan (*gradient*) pada dimensi ruang bobot dalam jaringan syaraf tiruan *multilayer networks*, proses pelatihan akan terus dilakukan secara terus menerus dan secara berulang hingga nilai *error* menjadi lebih minim daripada yang ditentukan di awal proses pelatihan. Penerapan metode serta penggunaan dari jaringan syaraf tiruan *backpropagation* tergolong algoritma pembelajaran/pelatihan yang memiliki sifat *supervised* dan menggunakan aturan pembelajaran dengan mengoreksi nilai error (Graupe, 2013). Berikut akan ditunjukkan arsitektur *Backpropagation* pada gambar 2.4.



Gambar 2.4 Arsitektur Backpropagation.

Menurut Fausset (1994) sebelum dilakukannya proses pelatihan pada terdapat parameter-parameter dalam jaringan syaraf *backpropagation* yang harus ditentukan terlebih dahulu sebelum memulai proses pelatihan yakni:

1. *Learning rate*
Nilai *alpha* atau *Learning rate* atau tingkat pembelajaran harus didefinisikan terlebih dahulu, memiliki nilai positif <1 .
2. *Error Tolerance*
Semakin sedikit nilai *error* yang ditentukan maka keakuratan yang dihasilkan oleh jaringan akan semakin baik, akan tetapi hal ini akan memakan waktu dan memperpanjang pelatihan.
3. *Maximum Epoch*
Untuk mencegah terjadinya perulangan tanpa akhir biasanya akan diberikan nilai *maximum epoch* di awal pelatihan.

2.5.1 Algoritma Proses Training pada Backpropagation

Berikut adalah tahap-tahap algoritma pelatihan *Backpropagation* untuk satu *hidden layer* (Fausset, 1994):

1. Inisialisasi bobot dan bias pada *backpropagation* dilakukan secara random atau dengan sembarang angka antara -0,5 hingga 0,5.
2. Tahap selanjutnya adalah lakukan tahap ini hingga kondisi berhenti belum terpenuhi. Proses menentukan kondisi berhenti adalah ketika mencapai iterasi maksimal yang ditentukan atau mencapai batas nilai *error* tertentu.
3. Akan dimulai proses dari *forward*, dalam setiap unit neuron pada unit input (X_i , $i = 1, \dots, n$) kemudian memperoleh sinyal input yaitu X_i dilanjutkan meneruskan sinyal *input* menuju dalam layer tersembunyi (*unit hidden* atau *hidden layer*),
4. Pada setiap dari unit tersembunyi ($Z_j, j = 1, \dots, p$) akan dilakukan proses penjumlahan sinyal *input* beserta bobot dan biasnya.

$$Zin_j = V_{0j} + \sum_{i=0}^n x_i v_{ij} \quad (2.1)$$

Diteruskan dengan perhitungan fungsi aktivasi akan didapatkan hasil dari sinyal keluaran (*output*) dari neuron unit tersembunyi. Fungsi aktivasi dapat dilakukan dengan persamaan 2.2 berikut:

$$Z_j = f(Z in_j) \quad (2.2)$$

5. Dalam setiap unit dari lapisan output ($Y_k, k = 1, \dots, m$) akan menjumlahkan sinyal-sinyal dari lapisan pada unit tersembunyi dengan bobot beserta bias.

$$Yin_k = W_{0k} + \sum_{i=0}^n Z_j w_{jk} \quad (2.3)$$

Dilanjutkna dengan menggunakan perhitungan rumus fungsi aktivasi untuk menghitung sinyal *output* dari unit pada *output*.

$$Y_k = f(Y in_k) \quad (2.4)$$

Kemudian akan dilanjutkan tahap selanjutnya yaitu pada proses *backward*.

6. Pada masing-masing unit dalam lapisan *output* ($Y_k, k = 1, \dots, m$) akan mendapat pola target output yang berkaitan dengan pola pada proses pembelajaran. Diantara target output dengan output yang dihasilkan melalui evaluasi akan dihitung nilai error. Jika belum terpenuhi maka dilakukan perhitungan informasi *error* pada k dengan persamaan 2.5.

$$\delta_k = (t_k - Y_k) f'(Y in_k) \quad (2.5)$$

Persamaan di atas berguna untuk menghitung koreksi bobot pada ΔW_{jk} yang akan digunakan pada *update* bobot W_{jk} .

Dilanjutkan pada tahap koreksi bobot dari W_{jk} dengan persamaan di bawah ini:

$$\Delta W_{jk} = \alpha \delta_k Z_j \quad (2.6)$$

Akan dihitung pula koreksi bias dari W_{0k}

$$\Delta W_{0k} = \alpha \delta_k \quad (2.7)$$

Kemudian akan dilanjutkan dengan disebarkan informasi dari nilai *error* pada lapisan yang berada di bawahnya.

7. Pada masing-masing unit tersembunyi akan dijumlahkan bobot dari unit *output*. Perhitungan dilakukan dengan menghitung hasil kali dari informasi error dengan bobot.

$$in_j = \sum_{k=1}^m \delta_k w_{jk} \quad (2.8)$$

Selanjutnya, hasil perhitungan dari δin_j dikalikan dengan hasil dari fungsi aktivasi dari unit tersembunyi untuk memperoleh faktor dari informasi *error*

$$\delta_j = \delta in_j f'(Z in_j) \quad (2.9)$$

Akan dihitung koreksi dari bobot pada V_{ij}

$$\Delta V_{ij} = \alpha \delta_j x_i \quad (2.10)$$

Kemudian dilanjutkan dengan menghitung koreksi bias dari V_{0j}

$$\Delta V_{0j} = \alpha \delta_j \quad (2.11)$$

Setelah selesai tahapan di atas maka dilanjutkan ke tahap *update* bobot.

8. Pada masing-masing dari unit pada lapisan *output* ($Y_k, k = 1, \dots, m$) memperbaiki (memperbaharui) bobot dan biasnya ($j = 0, \dots, p$)

$$W_{jk}(new) = W_{jk}(old) + \Delta W_{jk} \quad (2.12)$$

Dilanjutkan dengan masing-masing dari unit pada lapisan tersembunyi ($Z_j, j = 1, \dots, p$) memperbaiki bobot serta biasnya ($i = 0, \dots, n$)

$$V_{ij}(new) = V_{ij}(old) + \Delta V_{ij} \quad (2.13)$$

Kemudian dilanjutkan memeriksa kondisi berhenti (*stopping condition*).

9. Bila hasilnya *true*, maka proses training dapat berhenti.

Keterangan:

X_i = Unit input neuron ke- i

Y_k = Unit output neuron ke- k

Z_j = Unit hidden neuron ke- j

$f(x)$ = Fungsi aktivasi

α = *Learning rate*

n = Banyaknya neuron pada lapisan *input*

l = Banyaknya atau jumlah dari pasangan untuk data pelatihan

p = Banyaknya neuron lapisan unit *hidden*

v = Bobot lapisan pertama

w = Bobot lapisan kedua

k = Jumlah neuron pada *output layer*

x = Sinyal input yang digunakan

y = Sinyal untuk output (pada tingkat aktivasi dari output neuron)

z = Sinyal untuk output (pada tingkat aktivasi dari *hidden* neuron)

z_{in} = Sinyal pada input untuk *hidden* neuron

y_{in} = Sinyal pada input untuk output neuron

V_{ij} = Bobot yang terdapat di antara unit pada input ke- i dan unit hidden ke- j

V_{0j} = Nilai bias pada unit hidden ke- j

W_{jk} = Bobot yang terdapat di antara unit hidden ke- j dengan unit output ke- k

δ_k = Informasi nilai kesalahan (*error*) untuk mengkoreksi bobot W_{jk}

δ_j = Informasi nilai kesalahan (*error*) untuk mengkoreksi bobot V_{ij}

2.5.2 Fungsi Aktivasi

Fungsi aktivasi merupakan suatu persamaan matematis yang memungkinkan JST mengenali pola data serta memberikan hasil perhitungan dari data input yang belum pernah dipelajari. Menurut Haykin (1999) dalam jaringan syaraf tiruan terdapat berbagai macam fungsi aktivasi yang biasa digunakan, diantaranya:

1. Fungsi Sigmoid Biner

Fungsi aktivasi yang paling banyak dan umum digunakan adalah Fungsi Sigmoid Biner. Memiliki *range* nilai dari 0 hingga 1.

$$f_1(x) = \frac{1}{1 + \exp^{-x}} \quad (2.14)$$

Turunan dari fungsi tersebut adalah:

$$f_1'(x) = f_1(x)(1 - f_1(x)) \quad (2.15)$$

2. Fungsi Sigmoid Bipolar

Pada fungsi aktivasi ini, *range* nilai yang dapat digunakan berkisar antara -1 hingga 1.

$$f_2(x) = 2f_1(x) - 1 \quad (2.16)$$

Turunan dari fungsi tersebut adalah:

$$f_2'(x) = \frac{1}{2}(1 + f_2(x))(1 - f_2(x)) \quad (2.17)$$

3. Fungsi Tangen Hiperbolik

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.18)$$

Turunan dari fungsi tersebut adalah:

$$\tanh'(x) = (1 + \tanh(x))(1 - \tanh(x)) \quad (2.19)$$

2.6 Mean Square Error (MSE)

Mean Squared Error (MSE) adalah metode yang berguna mengevaluasi metode peramalan. Selain MSE terdapat berbagai macam metode lain untuk mengevaluasi peramalan. MSE adalah salah satu metode alternatif yang dapat digunakan dalam suatu metode peramalan (Yuniastari & Wirawan, 2014). Perhitungan nilai *error* berguna untuk melihat apakah peramalan dapat berjalan sesuai dan baik ataupun

digunakan untuk membandingkan prediksi/peramalan. MSE atau dapat disebut juga nilai tengah galat kuadrat bekerja dengan cara masing-masing nilai *error* atau hasil sisa dikuadratkan. Lalu dilakukan proses penjumlahan dengan nilai jumlah hasil observasi. Berikut merupakan rumus dari MSE (Bumi, 2016):

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - y'_t)^2 \quad (2.20)$$

Keterangan:

n = jumlah data yang di uji

y_t = nilai target awal

y'_t = nilai y pada unit output hasil perhitungan

MSE sangat tepat digunakan dalam memperoleh gambaran terhadap seberapa tepatnya model yang sedang dibuat. Memperkecil nilai MSE berarti pula memperkecil variasi model peramalan. Model bervariasi dengan jumlah minim memiliki keunggulan yakni hasil yang diberikan mampu lebih konsisten (tidak-berubah-ubah) untuk keseluruhan data input apabila kita lakukan perbandingan dengan model dengan variasi yang lebih besar.

2.7 Normalisasi dan Denormalisasi

Normalisasi dilakukan untuk menghasilkan struktur data dengan kondisi normal. Menurut Julpan, Nababan & Zarlis (2015) normalisasi data bertujuan memperoleh kesesuaian nilai data dengan jarak (cakupan) pada fungsi aktivasi yang telah dipergunakan dan ditetapkan dalam jaringan. Struktur data yang normal adalah struktur data yang dapat memenuhi kondisi tertentu. Hal ini dilakukan agar jaringan mampu mengenali data-data yang akan digunakan sebagai inputan bobot-bobotnya dan data tersebut harus mempunyai nilai yang sesuai berdasarkan pada fungsi aktivasi yang ditetapkan di awal. Pada penelitian ini akan digunakan proses normalisasi menggunakan persamaan 2.21 berikut (Wanto & Windarto 2017):

$$x' = \frac{0.8(x-\min)}{\max-\min} + 0.1 \quad (2.21)$$

Keterangan:

x' = Nilai hasil perhitungan normalisasi

x = Nilai sebenarnya (*real*) pada data yang akan dinormalisasi

\min = nilai minimum pada dataset

\max = nilai maximum pada dataset

Kemudian setelah seluruh proses dari sistem telah selesai, data akan dirubah kembali menjadi nilai aslinya. Proses merubah data menjadi nilai asli dinamakan

proses denormalisasi. Persamaan 2.22 dapat digunakan pada proses denormalisasi tersebut.

$$x = \frac{(max-min) (hasil\ uji-0.1)}{0.8} + min \quad (2.22)$$

Keterangan:

x = nilai asli pada data yang akan dinormalisasi

Min = nilai minimum pada dataset

Max = nilai maximum pada dataset

Hasil uji = nilai yang didapatkan dari hasil pengujian (y)

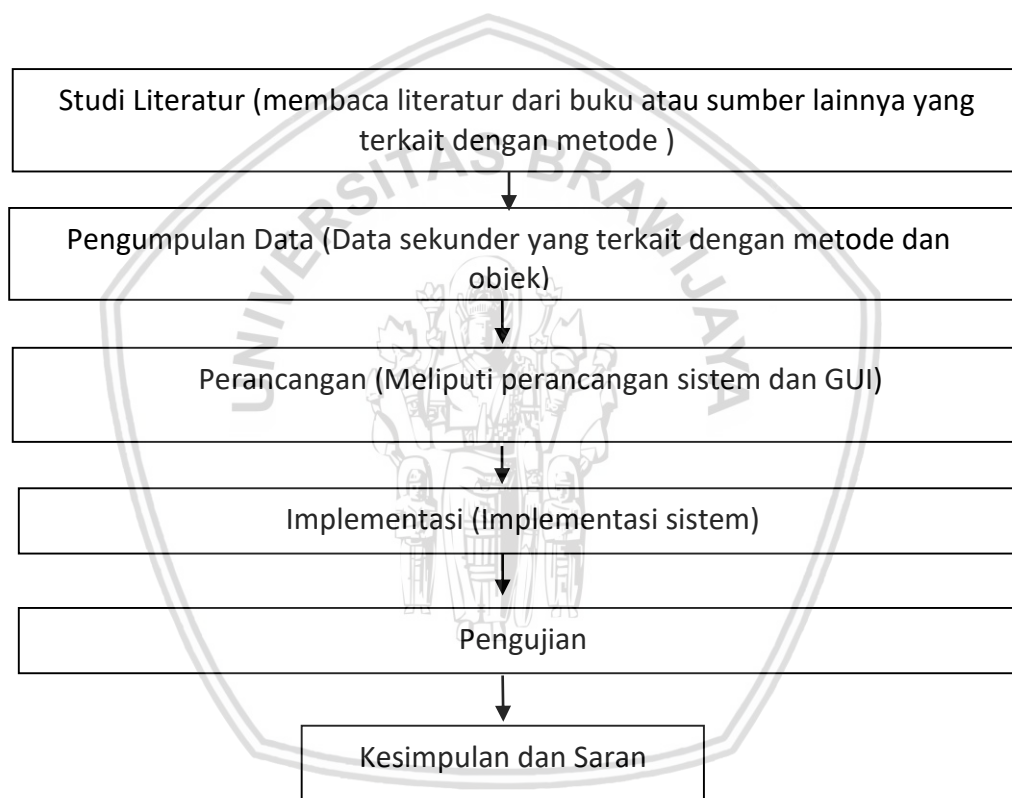
2.8 Stop Condition

Stop condition adalah suatu kriteria tertentu yang telah ditetapkan di awal pada proses pelatihan jaringan syaraf tiruan. Pada metode *Backpropagation*, proses *training* akan dilakukan secara berulang-ulang dan akan berhenti apabila *stop condition* (kondisi berhenti) telah terpenuhi. Stop condition pada metode *Backpropagation* dapat ditunjukkan seperti di bawah ini (Kholis & Alam 2016):

- Telah mencapai batas error tertentu. Perhitungan nilai *error* yang digunakan adalah menggunakan MSE (*Mean Square Error*).
- Mencapai batas iterasi maksimum atau epoch yang telah di tentukan di awal proses pelatihan.

BAB 3 METODOLOGI

Pada bab 3 ini akan dibahas mengenai metodologi penelitian. Dalam bab ini dijelaskan mengenai proses tahapan pada penelitian “Implementasi Metode *Backpropagation* Untuk Prediksi Harga Batu Bara” ini. Dalam metodologi penelitian akan dilakukan beberapa tahap yaitu melakukan studi literatur, tahapan pengumpulan data, proses analisis kebutuhan, perancangan, implementasi, pengujian serta kesimpulan. Tahapan yang akan dilakukan tersebut ditunjukkan pada Gambar 3.1 berikut.



Gambar 3.1 Diagram Alir Metodologi

3.1 Studi Literatur

Pada tahap ini, akan dilakukan studi literatur atau studi pustaka guna menunjang pembuatan sistem. Tahap ini berfungsi dan bertujuan untuk menunjang penulisan dan pembuatan skripsi ini. Studi literatur dapat diperoleh dari berbagai sumber *valid* dan terpercaya seperti jurnal, berbagai karya tulis ilmiah maupun penelitian yang telah dilakukan sebelumnya. Beberapa literatur yang akan dipelajari antara lain adalah:

1. Prediksi / peramalan (*prediction/forecasting*)
2. Harga Batu Bara
3. Metode *Backpropagation*.

3.2 Pengumpulan Data

Tahap pengumpulan data adalah suatu proses dalam memperoleh dan proses pengumpulan dari data yang akan digunakan pada penelitian ini. Berikut adalah data yang akan digunakan pada proses pembuatan penelitian ini:

1. Data yang didapatkan berupa data harga batu bara dimulai dari bulan Januari tahun 2009 hingga bulan September tahun 2017 dengan merk dagang Gunung Bayan I.
2. Data harga batu bara yang akan digunakan bersifat *time series* yaitu Harga Patokan Batu Bara (HPB) yang diperoleh dari laman resmi Ditjen Minerba, Kementerian ESDM.
3. Pada proses pengumpulan data, objek dari permasalahan diperoleh berdasarkan hasil dari proses wawancara yang dilakukan dengan narasumber. Narasumber dari penelitian ini adalah salah satu karyawan dari sebuah perusahaan batu bara di Indonesia.
4. Dalam penelitian yang akan dilakukan, data akan dibagi menjadi dua bagian yaitu data pelatihan (*training*) dan data uji (*testing*).

Tabel 3.1 Data Harga Patokan Batu Bara Gunung Bayan I

Tanggal	HPB Gunung Bayan I (USD/ton)
Januari-2009	84.64
Februari-2009	87.53
Maret-2009	80.73
April-2009	67.74
Mei-2009	67.46
Juni-2009	68.57
Juli-2009	76.6
Agustus-2009	76.83
September-2009	75.7
Oktober-2009	71.63

3.3 Perancangan

Pada tahap perancangan akan dilakukan perancangan sistem yang bertujuan agar sistem dapat tercipta dengan baik dalam segala aspek yang meliputinya.

Perancangan dilakukan untuk mempermudah dalam hal implementasi maupun pengujian. Dalam tahap untuk mengoptimalkan suatu permasalahan akan dijelaskan dalam beberapa tahap diantaranya adalah:

1. Perancangan dari proses manualisasi sistem yang bertujuan untuk menuntaskan permasalahan yaitu dengan cara membuat sebuah rancangan studi masalah perhitungan dengan metode *Backpropagation* secara sederhana dengan data yang akan digunakan pada implementasi sistem.
2. Perancangan proses diagram alir pada sistem.
3. Perancangan UI (*User Interface*) / antarmuka untuk memudahkan tahap implementasi dari sistem berupa perancangan dari model *user interface* yang akan digunakan.
4. Perancangan pengujian yang akan dilakukan setelah sistem selesai di implementasikan.

3.4 Implementasi

Implementasi sistem dilakukan dalam bentuk kode program menggunakan bahasa pemrograman java dan menggunakan aplikasi Netbeans sebagai lingkungan pengembangannya. Implementasi yang dilakukan meliputi:

1. Pembuatan *user interface* atau antarmuka pengguna yaitu berupa halaman-halaman antarmuka Java.
2. Memasukkan (*input*) data hasil akuisisi pengetahuan untuk dijadikan informasi yang bermanfaat serta berguna bagi sistem.
3. Penerapan metode *backpropagation* dalam kode program yang telah dibuat dalam bahasa Java.

Pada proses implementasi metode *backpropagation* untuk prediksi harga batu bara ini, terdapat beberapa komponen yang dibutuhkan, diantaranya:

1. Perangkat Keras (Hardware)
 - a. Intel®Core™ i5-4210U CPU 2.7 GHz.
 - b. Laptop dengan RAM 4.00 GB dan 1 TB HDD.
2. Perangkat Lunak (Software)
 - a. Sistem Operasi Windows 8 64-bit
 - b. IDE Netbeans 8.1

3.5 Pengujian

Pengujian yang dilakukan yaitu dengan menguji sistem yakni dengan memasukkan nilai pada parameter *backpropagation* yang berbeda-beda. Tujuannya adalah untuk melihat dan mengetahui seberapa besar pengaruh dari parameter yang digunakan pada proses *training*. Setelah di peroleh nilai yang paling baik, nilai parameter tersebut akan digunakan dalam proses training dan nilai bobot akhir yang diperoleh akan digunakan dalam prediksi menggunakan data uji.

3.6 Penarikan Kesimpulan dan Saran

Kesimpulan berisi tahap yang telah dilalui dan hasil yang dicapai setelah melalui tahap-tahap sebelumnya. Saran berguna untuk memperbaiki kesalahan agar penelitian selanjutnya dapat membangun dan menyempurnakan hasil dari penelitian ini.

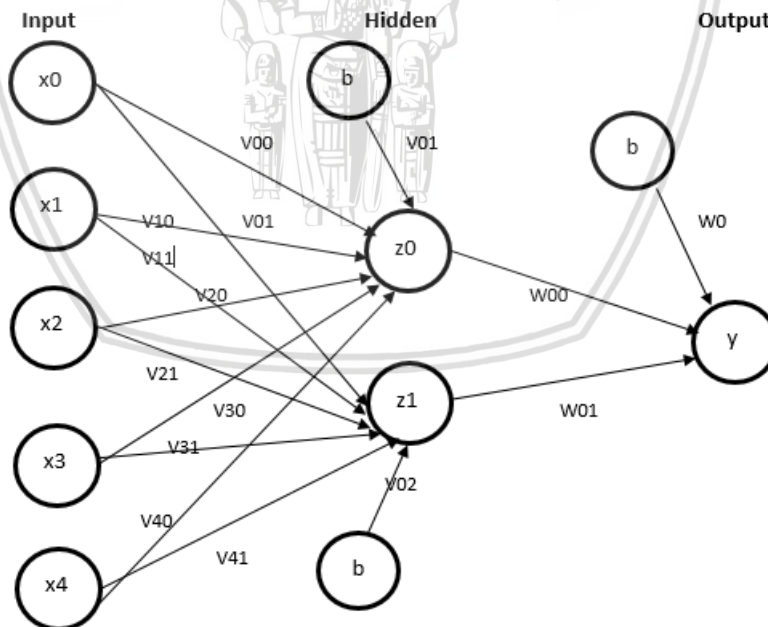


BAB 4 PERANCANGAN

Pada bab 4 ini yaitu bab perancangan akan ditunjukkan mengenai perancangan pada sistem Implementasi Metode *Backpropagation* untuk Prediksi Harga Batu Bara. Perancangan pada bab ini terdiri dari penjabaran deskripsi umum sistem, tahapan diagram alir sistem, proses perhitungan manual, perancangan antarmuka dan perancangan pengujian.

1.1 Deskripsi Umum Sistem

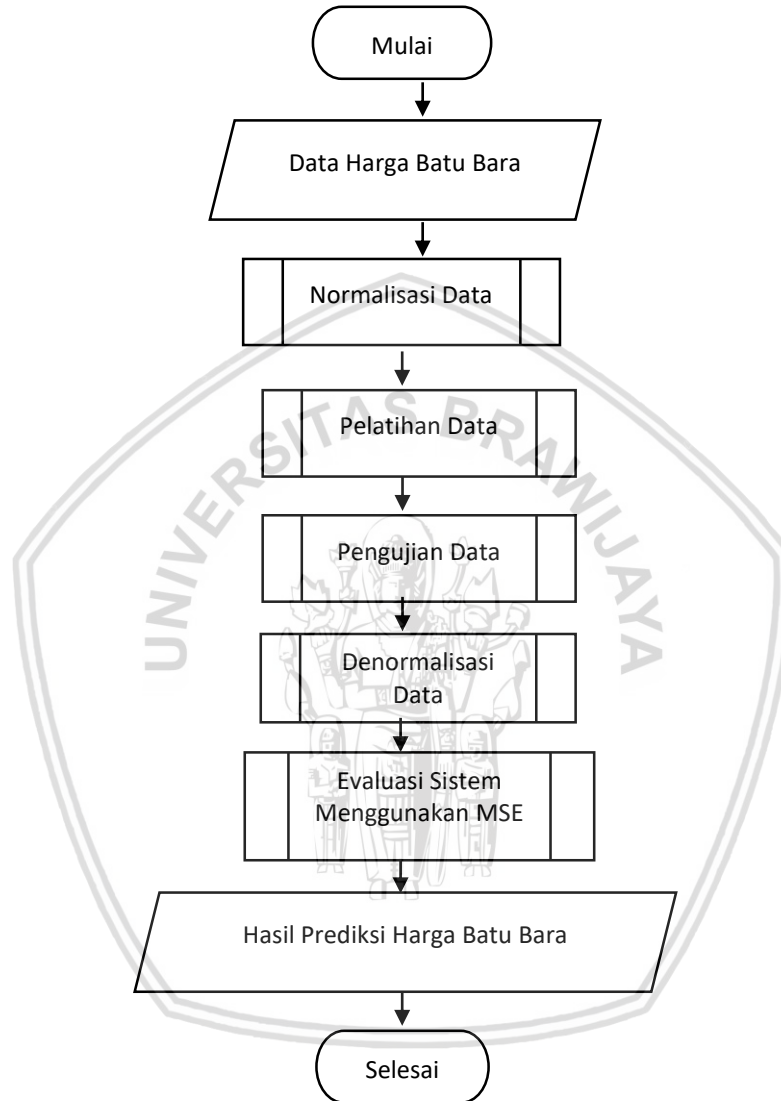
Sistem yang akan dibangun pada penelitian ini adalah sebuah sistem prediksi harga yang berbasis desktop. Masukan yang digunakan adalah data harga batu bara merk dagang Gunung Bayan I dari bulan Januari tahun 2009 hingga bulan September tahun 2017. Proses perhitungan yang dilakukan oleh sistem menggunakan metode *backpropagation*. Pada sistem yang akan dibangun, *user* dapat memasukkan nilai *learning rate* (α) yang akan digunakan. Selain itu jumlah fitur atau dapat dikatakan sebagai input layer, jumlah neuron pada *hidden layer*, nilai *max error* dan jumlah iterasi dapat ditentukan sendiri oleh *user*. Output yang dihasilkan adalah 1 neuron output yakni prediksi harga pada bulan selanjutnya. Berikut Gambar 4.1 menunjukkan arsitektur jaringan syaraf tiruan *backpropagation* dengan 5 neuron *input layer* X_i , $i \in \{0, 1, 2, 3, 4\}$ 2 neuron *hidden layer* Z_j , $j \in \{0, 1\}$ dan 1 neuron *output layer* (y).



Gambar 4.1 Arsitektur Jaringan Syaraf Tiruan *Backpropagation*

1.2 Diagram Alir Sistem

Pada gambar 4.2 akan menunjukkan tahapan-tahapan pada sistem secara umum dalam memprediksi harga batu bara menggunakan metode *backpropagation*.



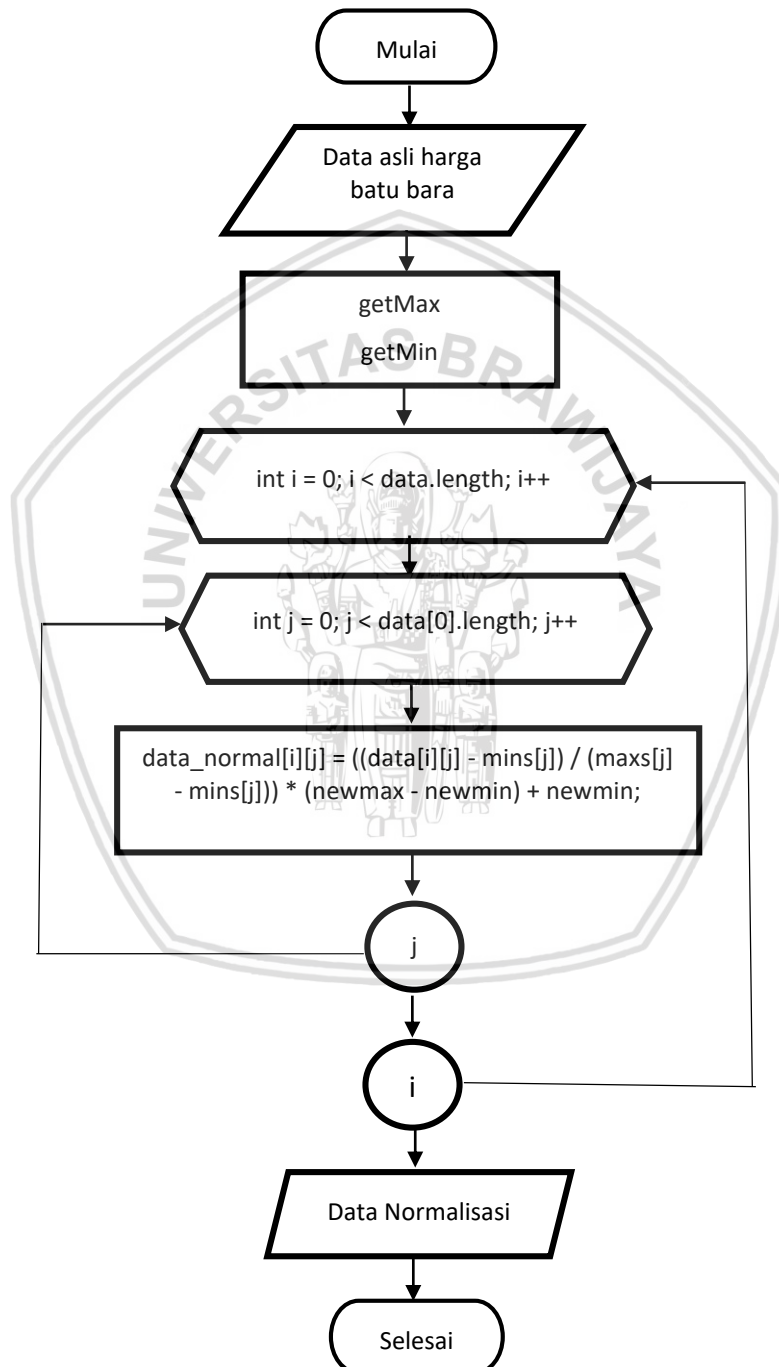
Gambar 4.2 Diagram Alir Sistem

Berikut adalah penjelasan tentang proses yang akan dilakukan dalam prediksi harga batu bara menggunakan metode *backpropagation*.

1. Input data harga batu bara yang hendak digunakan dalam proses prediksi. Data yang dimasukkan adalah data harga batu bara yang telah dipilih.
2. Kemudian data akan dilakukan proses normalisasi.
3. Selanjutnya proses pelatihan data atau *training*.
4. Langkah selanjutnya setelah dilakukan proses pelatihan adalah proses pengujian data (*testing*).

5. Dilakukan proses denormalisasi data, dilanjutkan dengan menghitung tingkat kesalahan menggunakan MSE.
6. Dihasilkan prediksi harga batu bara.

1.2.1 Diagram Alir Normalisasi Data



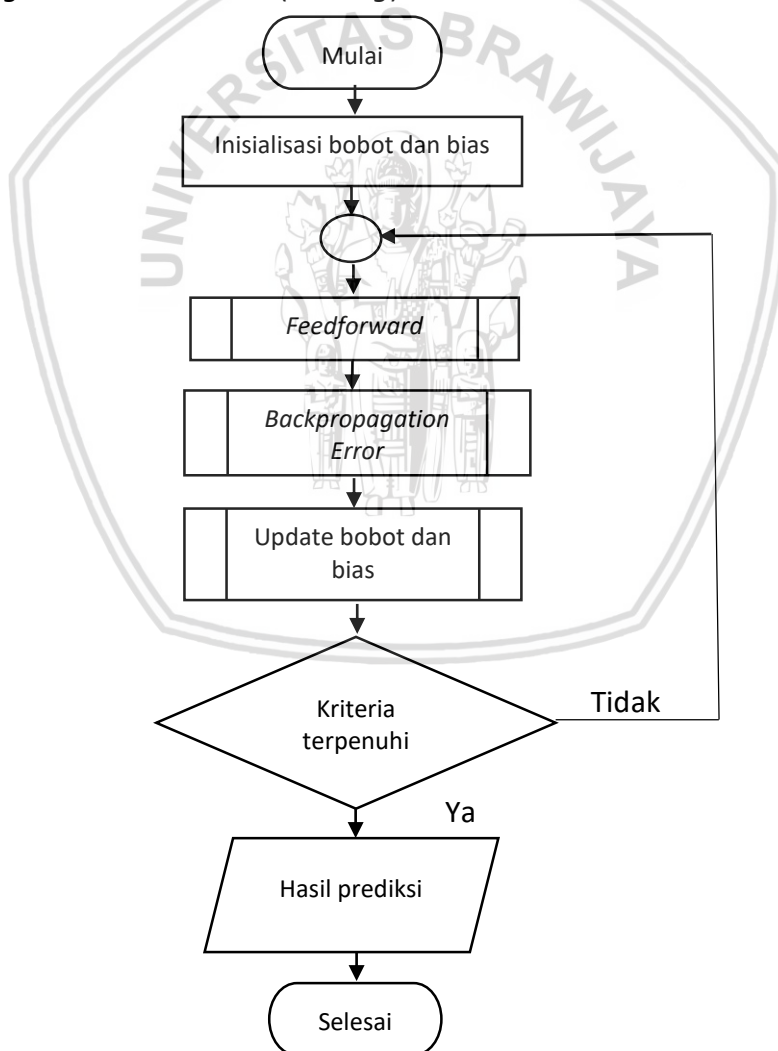
Gambar 4.3 Diagram Alir Proses Normalisasi

Pada Gambar 4.3 ditunjukkan alur proses normalisasi. Pada sistem yang akan dibuat, fungsi aktivasi yang digunakan ialah fungsi aktivasi sigmoid yang menghasilkan output berupa nilai antar 0 hingga 1 sehingga diperlukan normalisasi. Berikut akan dijelaskan mengenai diagram alir diatas:

1. *Input* berupa data asli harga batubara jenis Gunung Bayan I.
2. Dicari nilai min dan max dari tiap-tiap fitur
3. Proses perhitungan normalisasi menggunakan persamaan 2.21
4. *Output* berupa data normalisasi.

1.2.2 Diagram Alir Algoritma *Backpropagation*

Gambar 4.4 di bawah ini akan menunjukkan diagram alir dari algoritma *Backpropagation* secara umum (*training*).

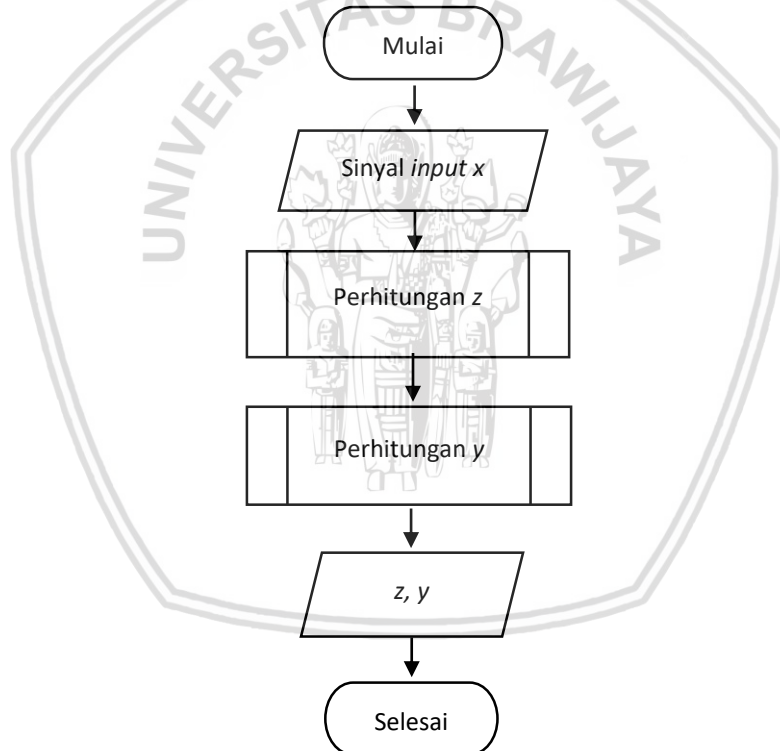


Gambar 4.4 Diagram Alir Algoritma *Backpropagation*

Berikut merupakan penjelasan dari tahapan-tahapan dalam diagram alir algoritma *Backpropagation*:

1. Inisialisasi bobot awal dan bias secara *random*
2. Dilanjutkan memasukkan nilai bobot dan bias yang telah ditentukan.
3. Dilakukan proses pengecekan pada nilai kriteria yang telah ditentukan di awal. Apabila terpenuhi maka akan *output* yang dihasilkan berupa hasil prediksi harga batu bara dan proses akan dihentikan. Jumlah iterasi merupakan kriteria yang dimaksud.
4. Apabila belum memenuhi kriteria yang dimaksudkan maka akan dilanjutkan pada tahap proses *feedforward*.
5. Kemudian dilakukan proses *backpropagation error*.
6. Lalu dilakukan tahap *update* bobot dan bias.

1.2.2.1 Diagram Alir Proses *Feedforward*



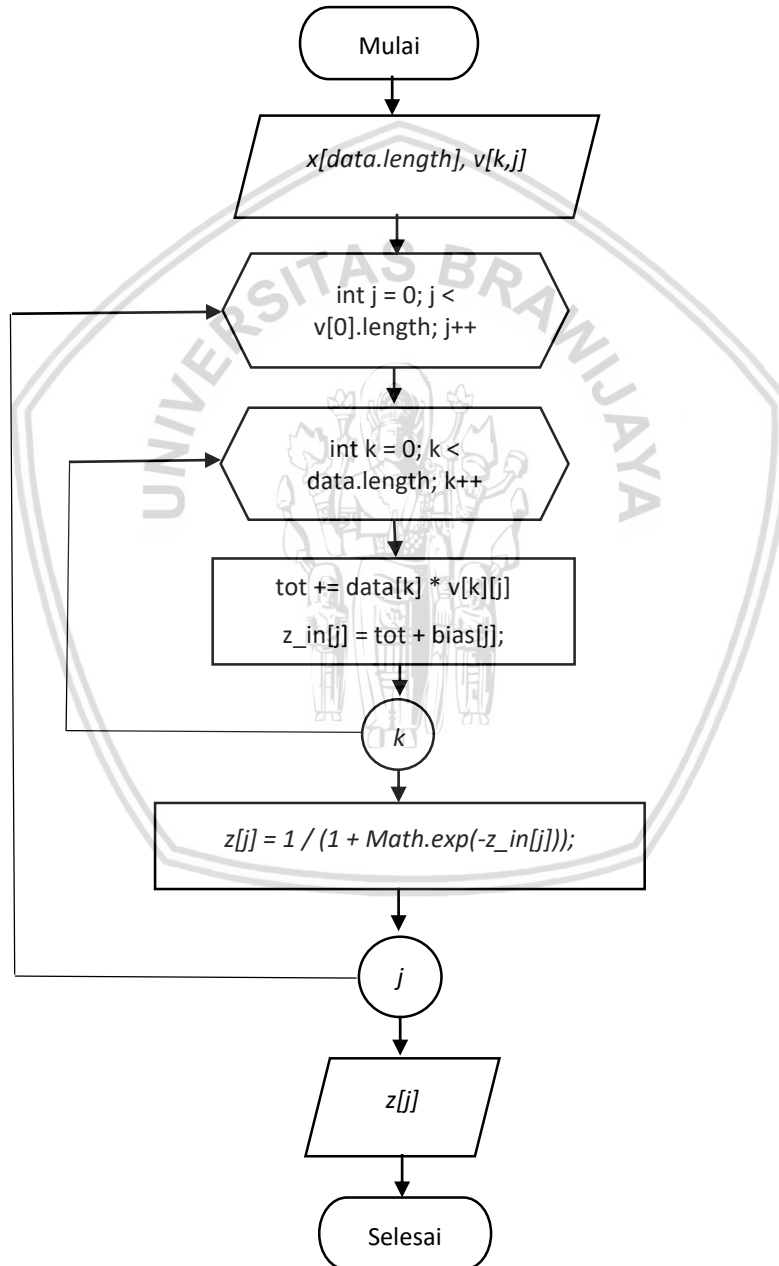
Gambar 4.5 Diagram Alir Proses *Feedforward*

Gambar 4.5 di atas menunjukkan diagram alir proses *feedforward*. Berikut adalah penjelasan mengenai tahap-tahap dari *feedforward*:

1. Nilai x pada gambar di atas adalah merepresentasikan harga batu bara pada bulan tertentu.

2. Kemudian sistem akan memproses perhitungan nilai *output* dari unit *hidden layer* (*z*) dengan menggunakan persamaan 2.1 dan 2.2.
3. Sistem akan memproses perhitungan nilai *output* dari unit *z* kemudian dilanjutkan menghitung nilai *y* (sinyal *output*) menggunakan persamaan 2.3 dan 2.4.
4. Mengeluarkan hasil berupa nilai *z* dan *y*.

1.2.2.2 Diagram Alir Perhitungan Nilai Z

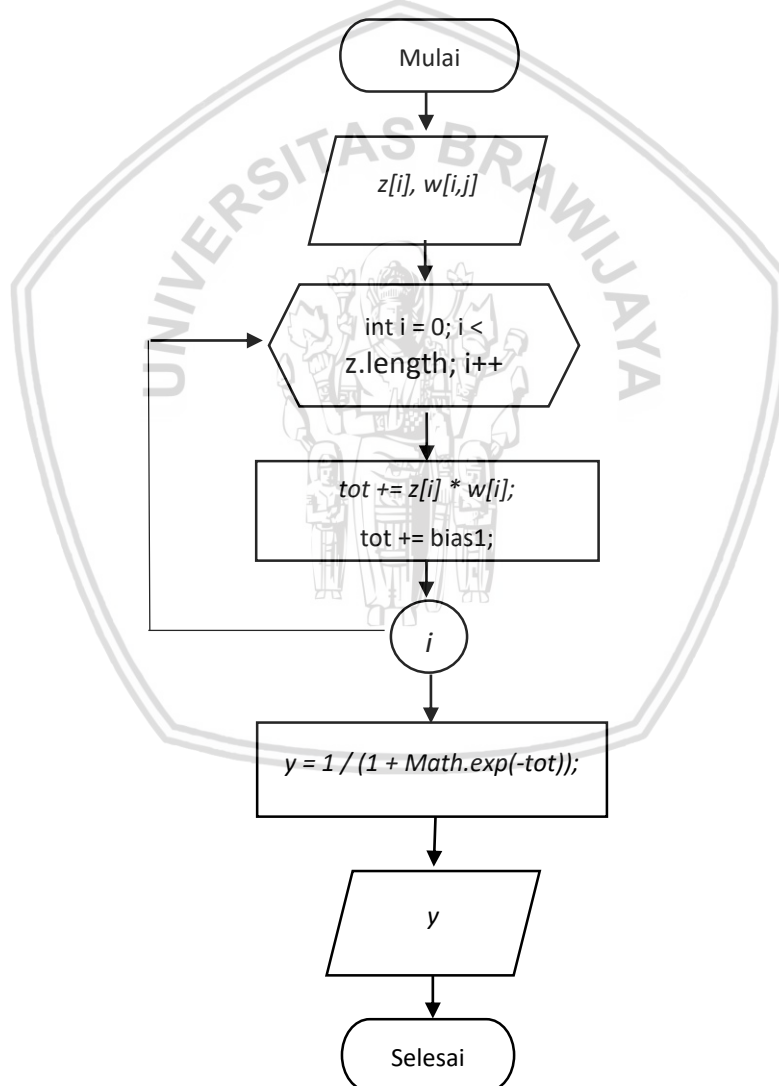


Gambar 4.6 Diagram Alir Perhitungan nilai z

Pada gambar 4.6 menunjukkan tahap dalam proses perhitungan output dari *hidden layer* (z). Proses perhitungan *hidden layer* (z) beserta fungsi aktivasinya dapat kita gunakan persamaan 2.1 dan 2.2. Berikut adalah penjelasan mengenai diagram alir diatas:

1. Masukan berupa *input* (x) dan *bias* (v).
2. Perulangan yang digunakan dalam proses perhitungan *input* yang telah berbobot (z_in), lalu melakukan proses perhitungan *output* dari *hidden layer* dengan fungsi aktivasi $z = f(z_in)$.
3. Menghasilkan *output* berupa nilai z .

1.2.2.3 Diagram Alir Perhitungan Nilai Y



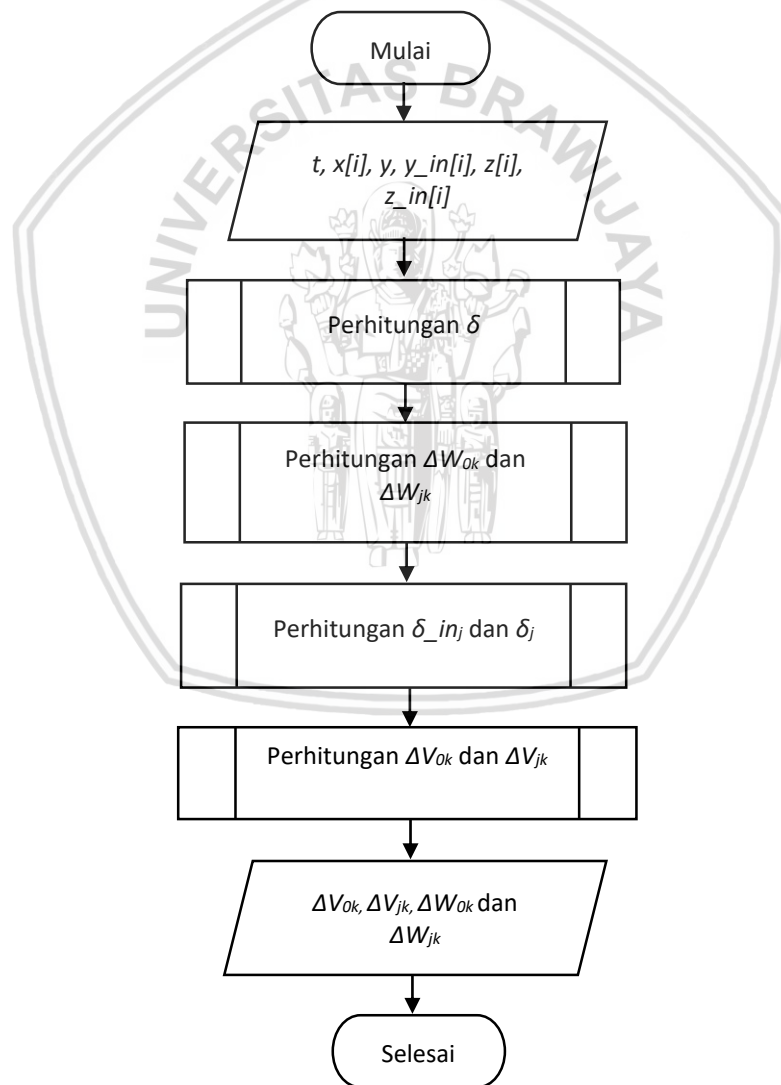
Gambar 4.7 Diagram Alir Perhitungan nilai y

Pada gambar 4.7 akan ditunjukkan mengenai tahapan yang terjadi pada proses penjumlahan untuk mencari nilai *output* dari unit *output layer* (y). Proses perhitungan

nilai y dapat menggunakan persamaan 2.3 dan 2.4. Berikut penjelasan dari diagram alir diatas:

1. Sistem memperoleh inputan berupa sinyal *output* yang diperoleh dari unit *hidden layer*(z) yang diperoleh dari proses sebelumnya serta bobot (w) sebagai *input*.
2. Melakukan perhitungan dengan melakukan perulangan guna menghitung sinyal dari *input* yang telah memiliki bobot (y_{ink}) dan melakukan perhitungan nilai dari sinyal *output* yang berasal dari *output layer* dengan menggunakan fungsi aktivasi $y = f(y_{in})$.
3. Mengeluarkan *output* berupa hasil perhitungan nilai y .

1.2.2.4 Diagram Alir Perhitungan *Backpropagation Error*

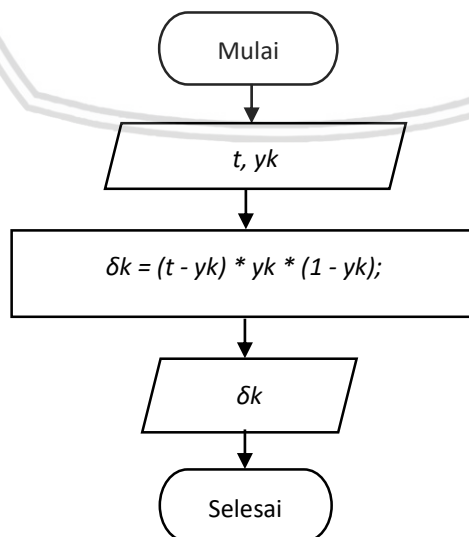


Gambar 4.8 Diagram Alir Perhitungan *Backpropagation Error*

Pada gambar 4.8 dijelaskan mengenai tahap demi tahap dalam proses *backpropagation error*. Berikut penjelasan mengenai tahapan dari diagram alir diatas:

1. Input yang akan diterima oleh sistem berupa t (target), $x[i]$ (harga batubara), y_k (nilai fungsi aktivasi dari sinyal output), $y_ink[i]$ (sinyal output), $z[i]$ (fungsi aktivasi hidden layer), $z_in[i]$ (unit hidden layer).
2. Kemudian akan dicari nilai δ (nilai error pada tiap output y_k), tujuannya adalah untuk mencari tingkat kesalahan yang terjadi antara *output* yang diperoleh dengan target. Proses perhitungan nilai δ dapat menggunakan persamaan 2.5.
3. Dilakukan perhitungan untuk mencari nilai ΔW_{ok} dan ΔW_{jk} , dengan tujuan untuk mencari tingkat dari nilai error (kesalahan) dari bobot dan bias (w). Nilai yang diperoleh pada tahap ini akan dipergunakan guna melakukan *update* bobot (w) di tahapan *update* (pembaruan) bobot dan bias. Proses perhitungan nilai ΔW_{ok} dan ΔW_{jk} dapat menggunakan persamaan 2.6.
4. Dilanjutkan dengan memproses perhitungan δ_in_j (nilai error pada unit *hidden layer*) dan δ_j (nilai error dari setiap unit output *hidden layer* atau Z_j). Proses perhitungan nilai δ_in_j dan δ_j dapat kita gunakan persamaan 2.8 dan 2.9.
5. Dilakukan perhitungan oleh sistem untuk mencari nilai ΔV_{ok} (koreksi bias) dan ΔV_{jk} (koreksi bobot), dengan tujuan untuk mencari tingkat dari nilai error (kesalahan) dari bobot dan bias (v). Nilai yang diperoleh pada tahap ini dipergunakan guna melakukan *update* bobot (v) di tahapan *update* (pembaruan) bobot dan bias. Proses perhitungan nilai ΔV_{ok} dan ΔV_{jk} dapat menggunakan persamaan 2.10 dan 2.11.
6. Output yang dihasilkan berupa nilai ΔV_{ok} , ΔV_{jk} , ΔW_{ok} dan ΔW_{jk} .

1.2.2.5 Diagram Alir Perhitungan nilai δ_k

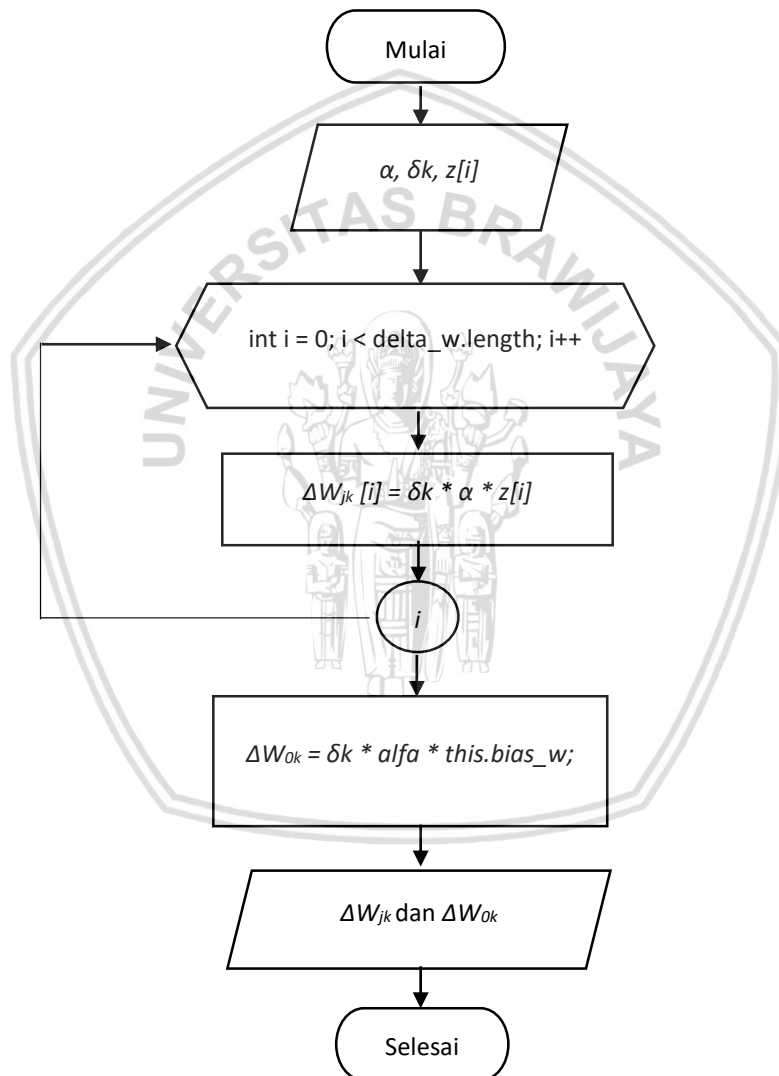


Gambar 4.9 Diagram Alir Perhitungan nilai δ_k

Gambar 4.9 di atas menunjukkan tahapan perhitungan untuk mencari nilai δk yaitu nilai error di tiap unit output y_k berdasarkan persamaan 2.5. Berikut penjelasan dari diagram alir di atas:

1. Sistem akan menerima *input* berupa nilai t (target) dan y_k (nilai fungsi aktivasi dari sinyal output).
2. Kemudian akan melakukan perhitungan nilai δk .
3. Sistem akan menghasilkan nilai *output* berupa nilai δk .

1.2.2.6 Diagram Alir Perhitungan Nilai ΔW_{jk} dan ΔW_{ok}

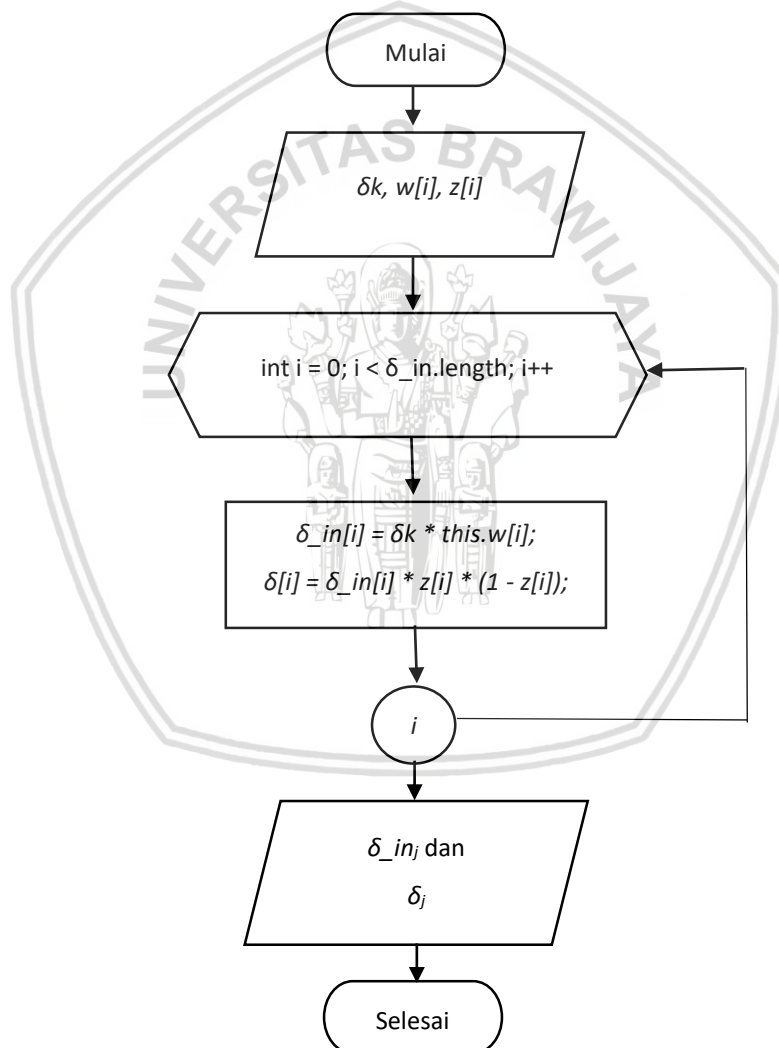


Gambar 4.10 Diagram Alir Perhitungan nilai ΔW_{jk} dan ΔW_{ok}

Gambar 4.10 menunjukkan diagram alir perhitungan nilai ΔW_{jk} dan ΔW_{ok} berdasarkan persamaan 2.6 dan 2.7. Berikut adalah penjelasan mengenai tahapannya:

1. *Input* yang diterima oleh sistem berupa nilai α (alpha), δk (nilai error di tiap unit output y_k), $z_j[i]$ (nilai hasil fungsi aktivasi dari perhitungan output *hidden layer*).
2. Kemudian sistem akan melakukan perhitungan secara perulangan untuk menghitung nilai dari ΔW_{jk} (koreksi bobot W_{jk}).
3. Setelah perhitungan perulangan selesai dilakukan, kemudian sistem akan melakukan perhitungan nilai ΔW_{0k} (koreksi bias W_{0k}).
4. Sistem akan menghasilkan output yaitu nilai ΔW_{jk} dan ΔW_{0k} .

1.2.2.7 Diagram Alir Perhitungan Nilai δ_{in_j} dan δ_j



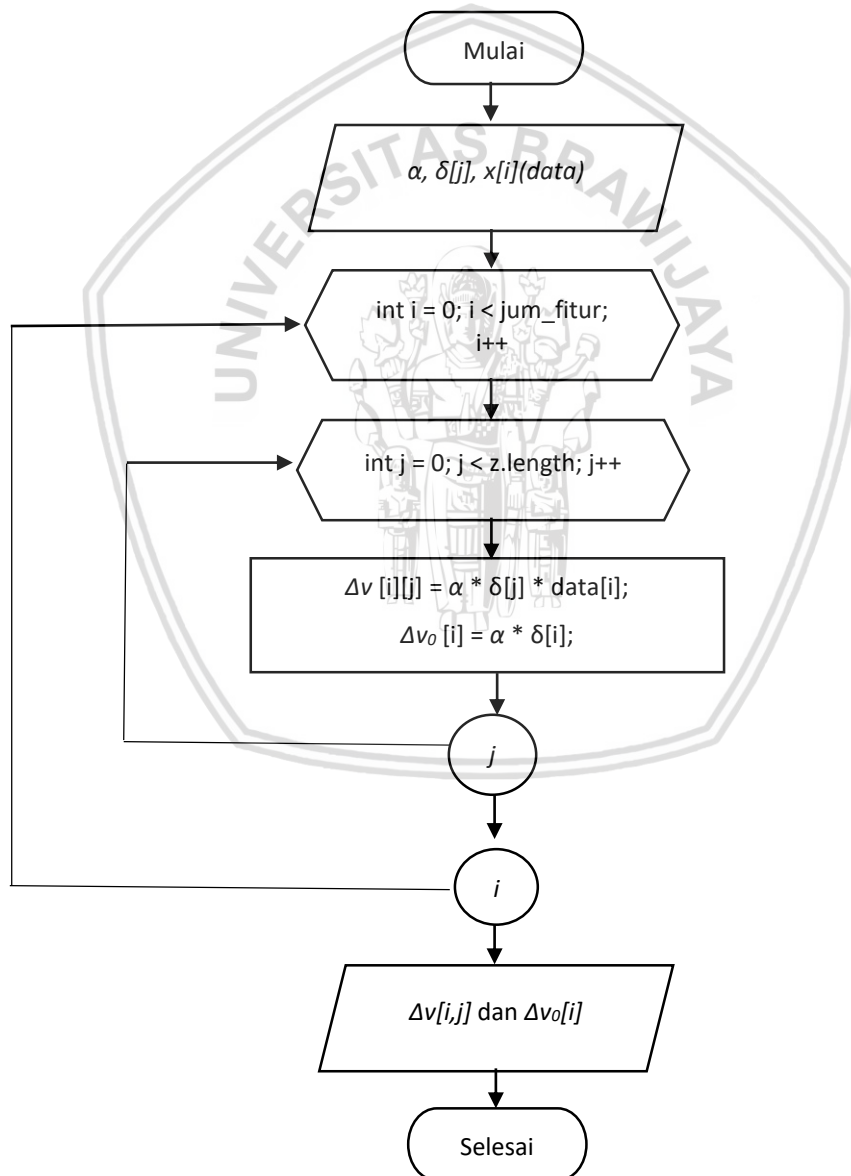
Gambar 4.11 Diagram Alir Perhitungan nilai δ_{in_j} dan δ_j

Gambar 4.11 di atas menunjukkan diagram alir dari tahap perhitungan nilai δ_{in_j} (nilai error pada unit *hidden layer*) dan δ_j (nilai error dari setiap unit output *hidden*)

layer atau Z_j) berdasarkan persamaan 2.8 dan 2.9. Berikut penjelasan mengenai tahapan-tahapan di atas:

1. Sistem akan menerima *input* yakni berupa nilai δ_k (nilai error di tiap unit output y_k), $w[i]$ (bobot), $zj[i]$ (nilai hasil fungsi aktivasi dari perhitungan output *hidden layer*).
2. Dilanjutkan sistem akan melakukan proses perhitungan perulangan nilai δ_{in_j} dan δ_j .
3. *Output* yang dihasilkan berupa nilai δ_{in_j} dan δ_j .

1.2.2.8 Diagram Alir Perhitungan Nilai ΔV_{ij} dan ΔV_{oj}

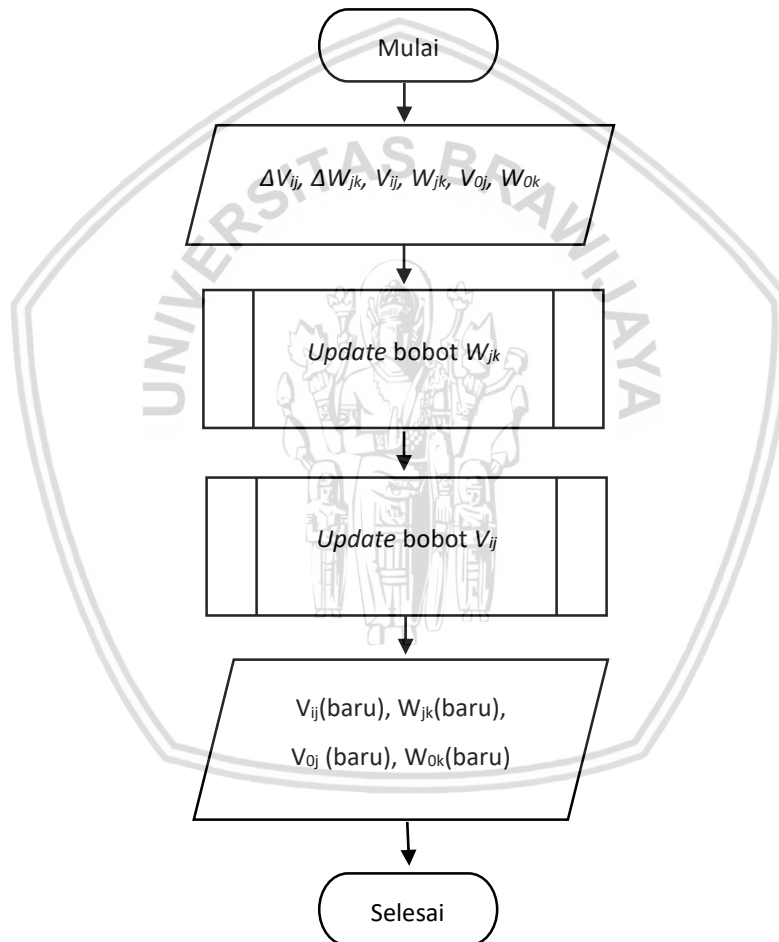


Gambar 4.12 Diagram Alir Perhitungan nilai ΔV_{ij} dan ΔV_{oj}

Gambar 4.12 di atas menunjukkan diagram alir dari tahap perhitungan nilai ΔV_{ij} dan ΔV_{oj} . Pada tahap perhitungan nilai ΔV_{ij} (koreksi bobot dari V_{ij}) dan ΔV_{ok} (koreksi bobot dari V_{oj}) dapat menggunakan persamaan 2.10 dan 2.11. Berikut penjelasan mengenai tahapan-tahapan di atas:

1. Sistem akan menerima *input* yakni berupa nilai α (alpha), $\delta[j]$ δ_j (nilai error dari setiap unit output *hidden layer* atau Z_j), $x[i]$ (nilai input)
2. Dilanjutkan sistem akan melakukan proses perhitungan nilai ΔV_{ij} dan ΔV_{oj}
3. *Output* yang dihasilkan berupa nilai ΔV_{ij} dan ΔV_{oj} .

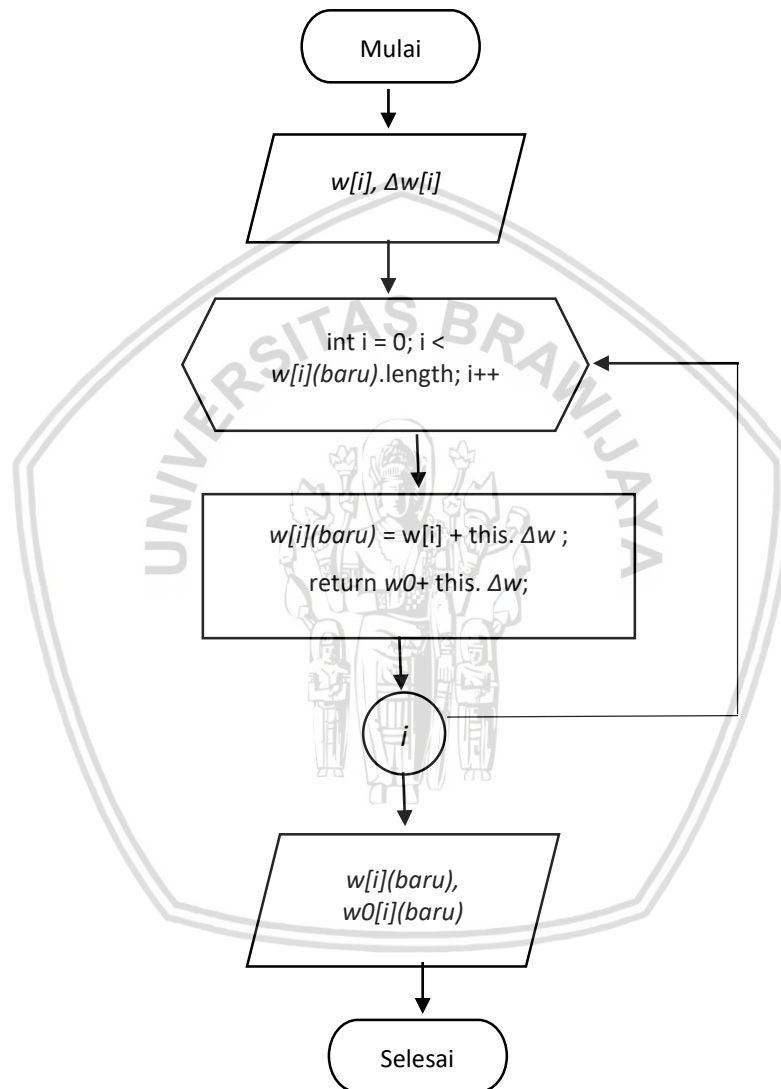
1.2.2.9 Diagram Alir Perhitungan *Update* Bobot dan Bias



Gambar 4.13 Diagram Alir Perhitungan *Update* Bobot dan Bias

Pada gambar 4.13 di atas ditunjukkan diagram alir dari proses update bobot dan bias secara garis besar. Berikut akan dijelaskan mengenai tahapan yang ada pada diagram alir di atas:

1. Pada awalnya sistem akan memperoleh inputan berupa nilai ΔV_{ij} , ΔW_{jk} , V_{ij} , W_{jk} , V_{0j} , dan W_{0k} .
2. Dilakukan perhitungan update bobot dari W_{jk} .
3. Dilakukan perhitungan update bobot dari V_{ij} .
4. Sistem akan menghasilkan output yaitu nilai dari $V_{ij}(\text{baru})$, $W_{jk}(\text{baru})$, $V_{0j}(\text{baru})$, dan $W_{0k}(\text{baru})$.

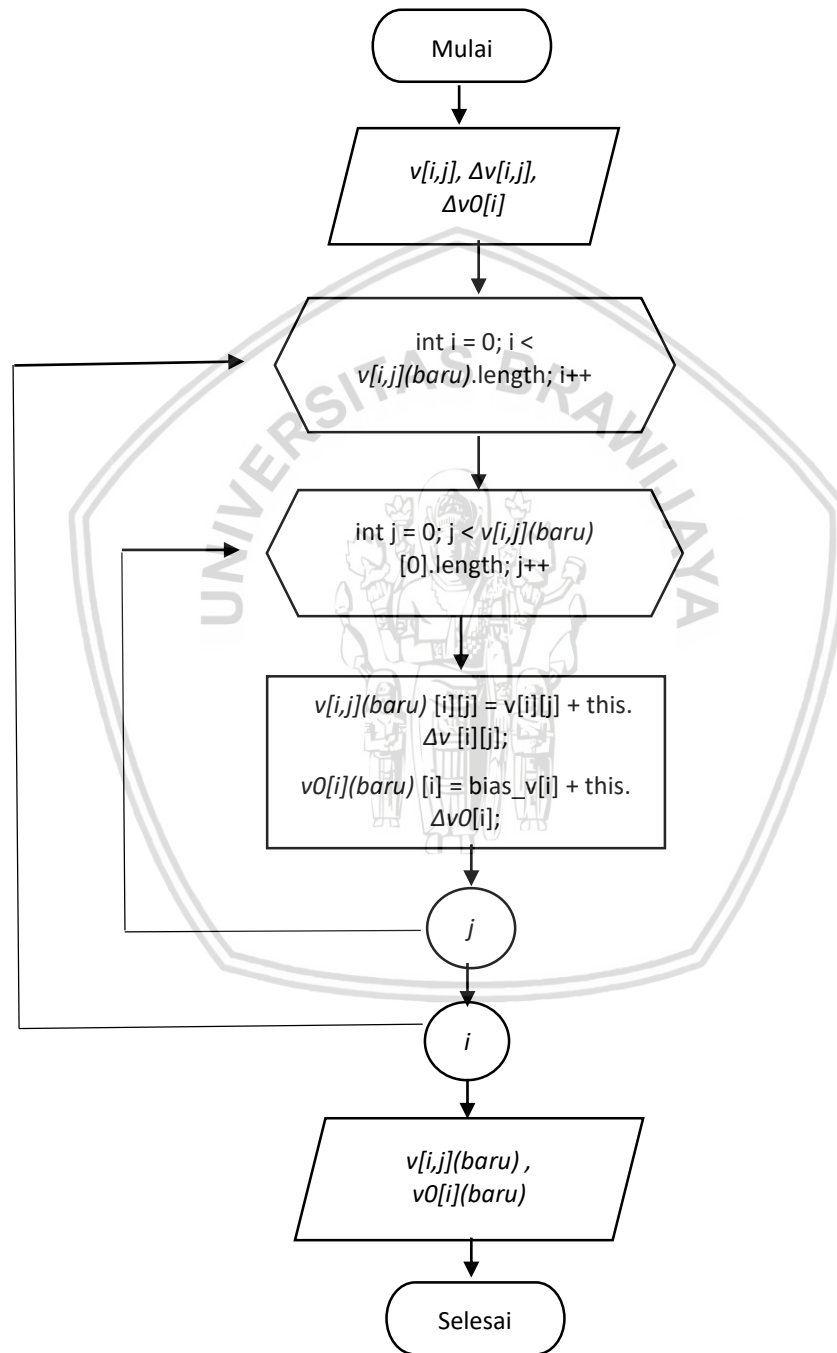


Gambar 4.14 Diagram Alir Perhitungan *Update* Bobot dan Bias (w_{jk} dan w_{0k})

Gambar 4.14 di atas menunjukkan diagram alir dari tahap perhitungan *Update* Bobot dan Bias (w_{jk} dan w_{0k}). Persamaan 2.12 digunakan dalam perhitungan ini. Berikut penjelasan mengenai tahapan-tahapan di atas:

1. *Input* yakni berupa nilai w_{jk} , Δw_{jk} akan diterima oleh sistem.

2. Dilanjutkan sistem akan melakukan proses perhitungan nilai pada *update* bobot (w_{jk}) dan bias (w_{0k}) *new* (bobot dan bias baru).
3. *Output* yang dihasilkan berupa hasil nilai bobot (w_{jk}) dan bias (w_{0k}) yang baru.

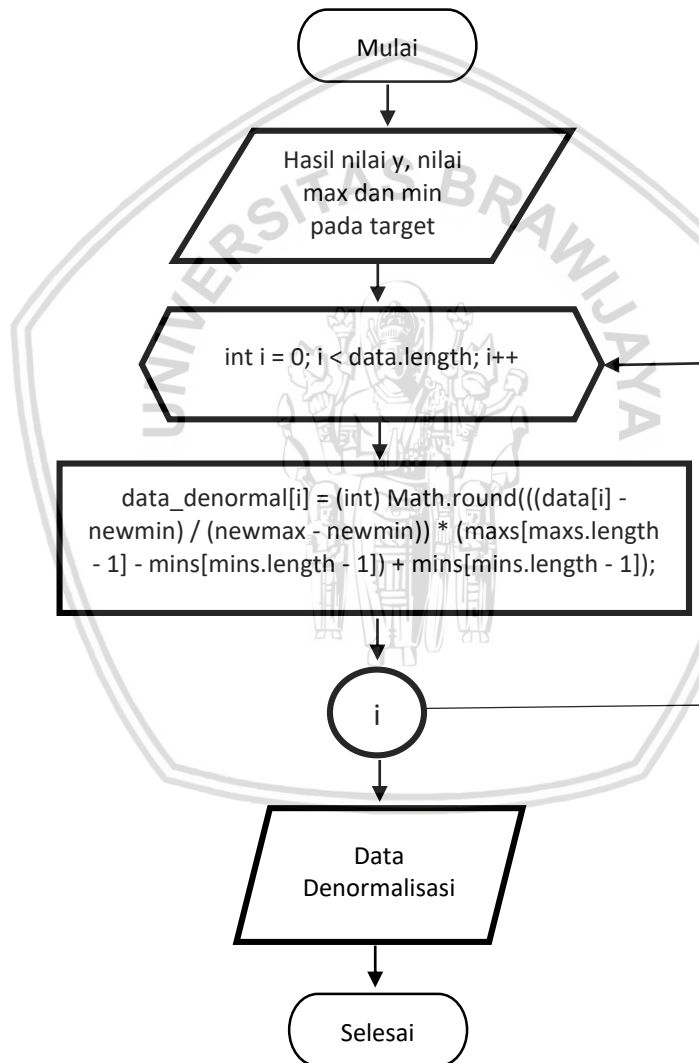


Gambar 4.15 Diagram Alir Perhitungan *Update* Bobot dan Bias (V_{ij} dan V_{0j})

Gambar 4.15 di atas menunjukkan diagram alir dari tahap perhitungan *Update* Bobot dan Bias (V_{ij} dan V_{0j}). Persamaan 2.12 digunakan dalam perhitungan ini. Berikut penjelasan mengenai tahapan-tahapan di atas:

1. *Input* yakni berupa nilai V_{ij} , ΔV_{0j} , ΔV_{ij} akan diterima oleh sistem.
2. Dilanjutkan sistem akan dilakukan proses perhitungan nilai pada *update* bobot (V_{ij}) dan bias (V_{0j}) yang baru.
3. *Output* yang dihasilkan berupa nilai bobot (V_{ij}) dan bias (V_{0j}) yang baru.

4.2.3 Diagram Alir Proses Denormalisasi Data

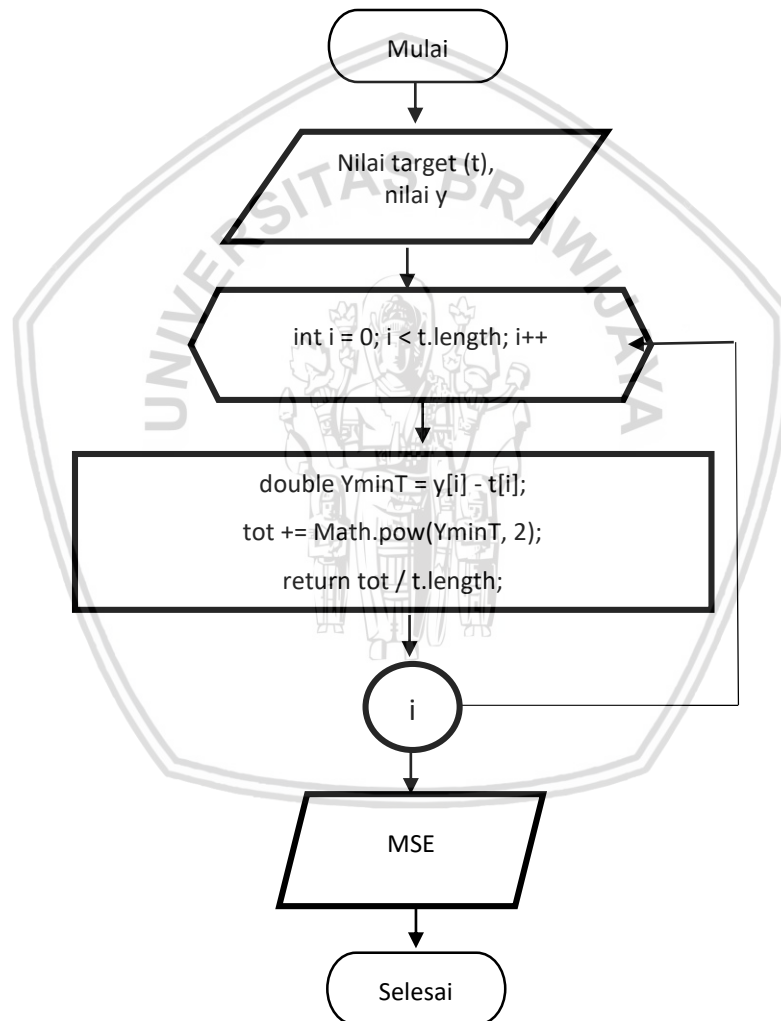


Gambar 4.16 Diagram Alir Proses Denormalisasi

Gambar 4.16 menunjukkan alur proses denormalisasi yang akan dilakukan oleh sistem. Denormalisasi berfungsi untuk mengubah data hasil dari normalisasi menjadi data asli atau data aktual. Berikut penjelasan dari diagram alir di atas:

1. *Input* berupa hasil nilai y , dan nilai max min pada target data aktual.
2. Proses perhitungan denormalisasi dilakukan menggunakan persamaan 2.22
3. *Output* berupa data hasil denormalisasi.

4.2.4 Diagram Alir Perhitungan MSE



Gambar 4.17 Diagram Alir Perhitungan MSE

MSE (*Mean Square Error*) digunakan untuk mengevaluasi hasil dari sistem dengan cara mengkuadratkan nilai *error* atau selisih di antara nilai *output* dan target.

Gambar 4.17 menunjukkan diagram alir perhitungan MSE. Berikut penjelasan mengenai diagram alir diatas:

1. *Input* berupa nilai target (t) dan nilai y akan diterima oleh sistem.
2. Dilanjutkan proses perhitungan MSE menggunakan persamaan 2.20 akan dilakukan oleh sistem.
3. *Output* berupa hasil MSE.

1.3 Perhitungan Manual

Pada perhitungan manual, digunakan data harga batu bara merk dagang Gunung Bayan I periode bulan Februari 2016 – November 2016 sebagai data latih (data *training*) dan periode bulan Januari 2017 – September 2017 sebagai data uji (data *testing*). Langkah awal dalam melakukan perhitungan manual ini adalah proses normalisasi data. Tahap ini bertujuan agar memudahkan dalam proses perhitungan nantinya.

Sistem prediksi harga batubara menggunakan metode *backpropagation* ini, terdapat struktur jaringan syaraf tiruan yang akan digunakan adalah sebagai berikut:

1. Neuron input layer: 5
2. Neuron hidden layer: 2
3. Neuron output: 1
4. Learning rate (α): 0.5

4.3.1 Penginputan Data

Pada perhitungan manual ini, data latih yang digunakan adalah data harga batu bara jenis Gunung Bayan I pada bulan Februari 2016 – November 2016. Harga batu bara yang digunakan dapat dilihat pada Tabel 4.1 berikut.

Tabel 4.1 Data Asli Harga Batu Bara

no	Bulan	Gunung Bayan I (USD/ton)
1	Februari-2016	54.36
2	Maret-2016	55.11
3	April-2016	55.87
4	Mei-2016	54.66
5	Juni-2016	55.32
6	Juli-2016	56.61
7	Agustus-2016	62.42
8	September-2016	68.45
9	Oktober-2016	74.01
10	November-2016	91.15

Kemudian data akan disesuaikan dengan arsitektur yang digunakan, yakni jumlah neuron input sebanyak 5 neuron. Berikut tabel data input harga batu bara setelah dirubah ditunjukkan pada Tabel 4.2.

Tabel 4.2 Data Input Harga Batu Bara

x1	x2	x3	x4	x5	Target	Kategori
54.36	55.11	55.87	54.66	55.32	56.61	Naik
55.11	55.87	54.66	55.32	56.61	62.42	Naik
55.87	54.66	55.32	56.61	62.42	68.45	Naik
54.66	55.32	56.61	62.42	68.45	74.01	Naik
55.32	56.61	62.42	68.45	74.01	91.15	Naik

Selanjutnya data akan dinormalisasi dengan persamaan di bawah ini:

$$x' = \frac{0.8(x - \min)}{\max - \min} + 0.1 \quad (2.21)$$

Berikut contoh perhitungannya:

$$x' = \frac{0.8(54.36 - 54.36)}{55.87 - 54.36} + 0.1$$

$$x' = \frac{0.8(0)}{1.51} + 0.1$$

$$x' = \frac{0}{1.51} + 0.1$$

$$x' = 0.1$$

Tabel 4.3 Data Input Hasil Normalisasi

x1	x2	x3	x4	x5	target
0.1	0.284615	0.224742	0.1	0.1	0.1
0.497351	0.59641	0.1	0.138289	0.155217	0.234569
0.9	0.1	0.168041	0.213125	0.403906	0.374233
0.25894	0.370769	0.301031	0.550181	0.662012	0.503011
0.608609	0.9	0.9	0.9	0.9	0.9

4.3.2 Inisialisasi Bobot Awal dan Bias

Inisialisasi bobot awal dan bias dilakukan secara *random* atau acak dengan memasukkan bilangan antara -0.5 sampai 0.5. Berikut dapat dilihat pada Tabel 4.4 untuk bobot V_{ij} , Tabel 4.5 untuk set bias V_0j dan Tabel 4.6 untuk bobot W_{jk}

Tabel 4.4 Bobot Vij

vij	j	
	1	2
x1	0.1267	-0.4657
x2	0.3349	0.3259
x3	-0.3912	0.1325
x4	0.2491	0.2655
x5	-0.4131	-0.1288

Tabel 4.5 Bobot V0j

v	i	
	1	2
0	0.2355	0.3777

Tabel 4.6 Bobot Wjk

Wjk	k
	1
0	0.25
1	-0.36
2	0.48

4.3.3 Feedforward

Tujuan dalam tahapan *feedforward* ini adalah untuk memperoleh nilai yang gunanya akan diteruskan pada lapisan selanjutnya. Menghitung *output hidden layer* dengan fungsi aktivasinya menggunakan persamaan 2.1 dan 2.2. Fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid biner dengan *range* (0,1). Berikut ini adalah contoh perhitungannya:

$$Zin_j = V_{0j} + \sum_{i=0}^n x_i v_{ij}$$

$$Zin_j = 0.2355 + ((0.1 * 0.1267) + (0.2846 * 0.3349) + (0.2247 * -0.391) + (0.1 * 0.2491) + (0.1 * -0.413))$$

$$Zin_j = 0.2355 + ((0.01267) + (0.0953) + (-0.0878) + (0.02491) + (-0.00413))$$

$$Zin_j = 0.2392$$

Tabel 4.7 Nilai Hasil Z_{in_j}

Z_{in1}	Z_{in2}
0.2392	0.4673

$$Z_j = f(Z_{in_j})$$

$$j = \frac{1}{1 + 2.71828183^{-0.2392}}$$

$$Z_j = 0.5595$$

Tabel 4.8 Nilai Hasil Z_j

$Z1=f(z_{in1})$	$Z2=f(z_{in2})$
0.5595	0.6147

Kemudian dilanjutkan dengan menghitung nilai y_{ink} menggunakan menggunakan persamaan 2.3 dan 2.4:

$$Y_{in_k} = W_{0k} + \sum_{i=0}^n Z_j W_{jk}$$

$$y_{ink} = 0.25 + ((-0.36 * 0.5595) + (0.48 * 0.6147))$$

$$y_{ink} = 0.25 + ((-0.2014) + 0.2950)$$

$$y_{ink} = 0.3436$$

Tabel 4.9 Nilai Hasil Y_{ink}

Y_{ink}
0.343658

$$Y_k = f(Y_{in_k})$$

$$y_k = \frac{1}{1 + 2.71828183^{-0.343658}}$$

$$y_k = 0.5850789$$

Tabel 4.10 Nilai Hasil y_k

y_k
0.5850789

Kemudian akan dilanjutkan pada tahap *backpropagation*.

4.3.4 Backpropagation

Pada tahap backpropagation, langkah awal yaitu dengan menghitung faktor δ unit *output layer* sesuai pada nilai *error* di tiap unit output y_k . Dapat digunakan persamaan 2.5.

$$\delta_k = (t_k - Y_k) f'(Y_{in_k})$$

$$\delta_k = (0.1 - 0.5850) * 0.5850(1 - 0.5850)$$

$$\delta_k = -0.11775$$

Tabel 4.11 Nilai Hasil δ_k

δ_{k1}
-0.117759

δ_k akan digunakan dalam perubahan bobot pada lapisan yang berada di bawahnya. Kemudian akan dilanjutkan dengan menghitung perubahan w_{jk} . Pada tahap ini berguna untuk mengubah bobot w_{jk} menggunakan *learning rate* (α) berdasarkan pada persamaan 2.6 dan 2.7.

$$\Delta w_{jk} = \alpha \delta_k Z_j$$

$$\Delta w_{jk} = (0.5 * -0.11775 * 0.5595)$$

$$\Delta w_{jk} = -0.0329$$

$$\Delta W_{0k} = \alpha \delta_k$$

$$\Delta W_{0k} = 0.5 * -0.11775$$

$$\Delta W_{0k} = -0.05888$$

Tabel 4.12 Nilai Hasil ΔW_{jk}

ΔW_{jk}	ΔW_0 /bias	ΔW_1	ΔW_2
	-0.05888	-0.03294	-0.0362

Kemudian akan dilanjutkan dengan melakukan perhitungan faktor δ pada unit *hidden layer* berdasarkan pada *error* yang ada pada setiap unit *hidden layer* Z_j berdasarkan pada persamaan 2.8 dan 2.9.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

$$\delta_{in_j} = -0.11775 * -0.36$$

$$\delta_{in_j} = 0.0424$$

Tabel 4.13 Nilai Hasil δ_{in_j}

δ_{in1}	δ_{in2}
0.0424	-0.0565

$$\delta_j = \delta_{in_j} f'(Z_{in_j})$$

$$\delta_j = 0.0424 * 0.5595(1 - 0.5595)$$

$$\delta_j = 0.0104$$

Tabel 4.14 Nilai Hasil δ_j

δ_1	δ_2
0.0104	-0.0278

Lalu hitung suku pada bobot V_{ij} untuk digunakan dalam mengubah bobot V_{ij} berdasarkan persamaan 2.10 dan hitung bias V_{ij} (V_{0j}) dengan persamaan 2.11.

$$\Delta V_{ij} = \alpha \delta_j x_i$$

$$\Delta V_{ij} = 0.0104 * 0.5 * 0.1$$

$$\Delta V_{ij} = 0.000522$$

$$\Delta V_{0j} = \alpha \delta_j$$

$$\Delta V_{0j} = 0.5 * 0.01044$$

$$\Delta V_{0j} = 0.005224$$

Tabel 4.15 Nilai Hasil ΔV_{ij}

j	Vij					V0j
	x1	x2	x3	x4	x5	Bias
1	0.00052241	0.00148685	0.001174069	0.0005224	0.00052241	0.005224
2	-0.0013944	-0.0039688	-0.003133914	-0.0013944	-0.0013944	-0.01394

4.3.5 Weight Update (Perubahan Bobot)

Pada tahap ini berfungsi untuk perhitungan bobot baru yang menuju pada unit *hidden layer* dan yang menuju ke lapisan *output layer*. Pada tahap ini digunakan persamaan 2.12 untuk $w_{jk}(\text{baru})$ dan 2.13 untuk $V_{ij}(\text{baru})$.

$$W_{jk}(\text{new}) = W_{jk}(\text{old}) + \Delta W_{jk}$$

$$w_{jk}(\text{baru}) = -0.36 + (-0.0329435)$$

$$w_{jk}(\text{baru}) = -0.392$$

$$w_{0k}(\text{baru}) = 0.25 + -0.0588$$

$$w_{0k}(\text{baru}) = 0.1912$$

Tabel 4.16 Nilai Bobot $w_{jk}(\text{baru})$

Wjk	j		
	(Bias) 0	1	2
1	0.1912	-0.3929	0.4438

Kemudian untuk *update* bobot baru yang menuju ke *hidden layer*:

$$V_{ij}(\text{new}) = V_{ij}(\text{old}) + \Delta V_{ij}$$

$$V_{0j}(\text{baru}) = 0.2355 + 0.005224$$

$$V_{0j}(\text{baru}) = 0.240724$$

$$V_{ij}(\text{baru}) = 0.000522 + 0.1267$$

$$V_{ij}(\text{baru}) = 0.12750$$

Tabel 4.17 Nilai Bobot $V_{ij}(\text{baru})$

1	j	Vij					
		x1	x2	x3	x4	x5	Bias
	1	0.1272224	0.336387	-0.39003	0.249622	-0.41258	0.240724
	2	-0.467094	0.321931	0.129366	0.264106	-0.13019	-0.00527

4.3.6 Pengujian

Pada proses pengujian data yang digunakan adalah data pada bulan Januari 2017 – September 2017. Data pada bulan tersebut digunakan sebagai data uji (*data testing*). Bobot akhir yang diperoleh dari hasil *training* atau proses pelatihan akan digunakan pada proses pengujian ini. Data input harga batu bara sebagai data uji dapat dilihat pada Tabel 4.17. Sedangkan untuk hasil dari pengujian dapat dilihat pada Tabel 4.18 berikut.

Tabel 4.17 Tabel Input Harga Batu Bara (Data Testing)

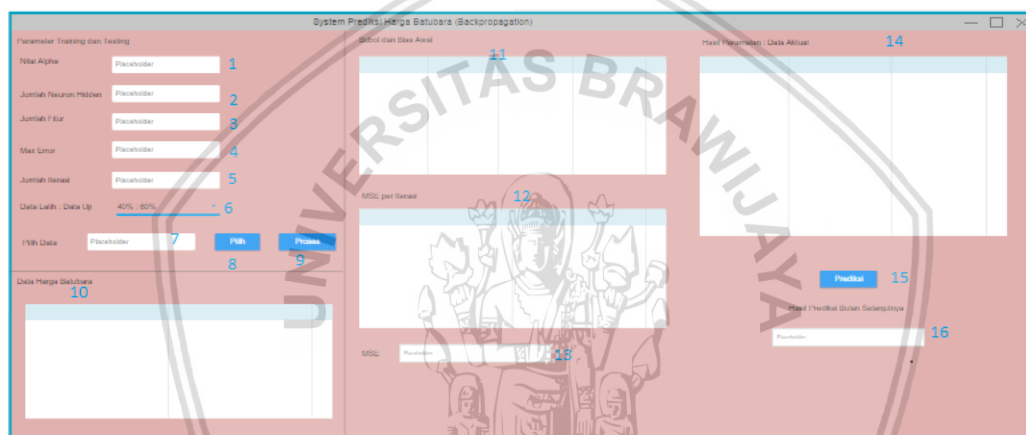
x1	x2	x3	x4	x5	target
109.35	92.6	89.45	87.91	88.57	89.98
92.6	89.45	87.91	88.57	89.98	80.94
89.45	87.91	88.57	89.98	80.94	84.72
87.91	88.57	89.98	80.94	84.72	90.15
88.57	89.98	80.94	84.72	90.15	98.88

Tabel 4.18 Hasil Pengujian

x1	x2	x3	x4	x5	Target	Hasil
109.35	92.6	89.45	87.91	88.57	89.98	91.29
92.6	89.45	87.91	88.57	89.98	80.94	91.48
89.45	87.91	88.57	89.98	80.94	84.72	91.42
87.91	88.57	89.98	80.94	84.72	90.15	91.51
88.57	89.98	80.94	84.72	90.15	98.88	91.35

1.4 Perancangan Antarmuka

Pada Gambar 4.15 akan ditunjukkan model perancangan dari sistem yang akan dibangun.



Gambar 4.15 Rancangan Antarmuka Sistem

Terdapat 3 grup dalam rancangan antarmuka sistem, yaitu grup yang berisi parameter training dan testing, grup hasil serta grup data harga batu bara. Pada masing-masing grup memiliki fungsi yang berbeda. Label diberikan 1 hingga 16 diberikan untuk mendefinisikan fungsi setiap grup. Berikut merupakan keterangan dari label pada Gambar 4.15:

1. *Text field* nilai *alpha* digunakan untuk menginputkan nilai *alpha*.
2. *Text field* jumlah neuron hidden digunakan untuk menginputkan jumlah hidden layer yang diinginkan.
3. *Text field* jumlah fitur digunakan untuk menginputkan jumlah fitur yang akan digunakan.
4. *Text field* max error digunakan untuk menginputkan jumlah max error yang akan digunakan pada sistem.
5. *Text field* jumlah iterasi digunakan untuk menginputkan jumlah iterasi yang akan digunakan.

6. *Menu dropdown* digunakan untuk menentukan pilihan *range* data latih dan data uji dari keseluruhan data.
7. *Text field* pilih data digunakan untuk memilih data yang akan digunakan oleh sistem.
8. *Button* pilih digunakan untuk mencari dan menginputkan data ke dalam sistem.
9. *Button* proses digunakan untuk memulai proses perhitungan setelah parameter dari *backpropagation* dan data di inputkan.
10. *Grid view* data harga batubara untuk menampilkan dataset yang digunakan.
11. *Grid view* bobot dan bias awal untuk menampilkan bobot dan bias awal yang ditentukan secara random.
12. *Grid view* MSE per iterasi digunakan untuk menampilkan hasil MSE yang diperoleh setiap iterasi.
13. *Text field* MSE digunakan untuk menampilkan hasil perhitungan MSE akhir setelah proses iterasi selesai.
14. *Grid view* hasil peramalan data actual digunakan untuk menampilkan data hasil prediksi.
15. *Button* prediksi digunakan untuk menampilkan hasil prediksi.
16. *Text Field* hasil prediksi bulan selanjutnya menampilkan hasil prediksi untuk bulan berikutnya.

1.5 Perancangan Pengujian

Pada tahap perancangan pengujian akan dilakukan setelah seluruh sistem ini selesai dibuat. Pengujian bertujuan agar dapat diketahui nilai parameter *backpropagation* terbaik yang digunakan pada sistem agar dapat diperoleh hasil prediksi dengan tingkat akurasi yang tinggi serta untuk melihat bagaimana pengaruh parameter dari *backpropagation* terhadap hasil prediksi. Parameter yang akan diuji meliputi:

1. Variasi jumlah fitur (neuron input) dan neuron *hidden*.
2. Nilai *learning rate* (α).
3. Variasi jumlah data *training*.
4. Variasi jumlah iterasi.

4.5.1 Pengujian Jumlah Neuron pada Input Layer dan Hidden Layer Terbaik

Pada pengujian jumlah neuron pada *input* dan *hidden layer* ini bertujuan untuk mencari arsitektur terbaik berdasarkan nilai MSE yang dihasilkan. Pengujian yang akan dilakukan dengan menggunakan 3, 5, 7, 10 dan 15 neuron pada *input layer* (jumlah fitur) dan 1 sampai 10 pada *hidden layer*. Pada Tabel 4.19 akan ditunjukkan tabel yang akan digunakan pada pengujian jumlah input *neuron* pada *input layer* dan *hidden layer* terbaik.

Tabel 4.19 Pengujian Jumlah Neuron pada *Input Layer* dan *Hidden Layer* Terbaik

Jumlah Neuron Hidden	Jumlah Neuron Input				
	3	5	7	10	15
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

4.5.2 Pengujian Nilai *Learning Rate* (α)

Pada pengujian nilai *learning rate* akan dilakukan pencarian nilai *learning rate* yang menghasilkan rata-rata MSE terendah. Pengujian akan dilakukan dengan menggunakan nilai *learning rate* dari 0,1 – 0,9. Arsitektur yang akan digunakan adalah pasangan jumlah neuron *input* dan *hidden layer* terbaik yang didapatkan dari pengujian jumlah neuron pada *input* dan *hidden layer* terbaik. Tabel 4.20 akan ditunjukkan tabel yang akan digunakan pada pengujian ini.

Tabel 4.20 Pengujian Jumlah *Learning Rate*

Jumlah Learning Rate	Nilai MSE Percobaan ke-i					Rata - Rata MSE
	1	2	3	4	5	
0,1						
0,2						
0,3						
0,4						
0,5						
0,6						
0,7						
0,8						
0,9						

4.5.3 Pengujian Variasi Jumlah Data *Training*

Tujuan dari pengujian ini adalah untuk mengetahui nilai rata-rata MSE terendah dari variasi jumlah data *training* yang telah ditentukan. Dari pengujian ini akan didapatkan variasi jumlah data *training* terbaik. Tabel 4.21 akan menunjukkan mengenai tabel pengujian variasi jumlah data latih (*training*).

Tabel 4.21 Pengujian Variasi Jumlah Data Training

Jumlah Data <i>Training</i>	Nilai MSE Percobaan ke-i					Rata - Rata MSE
	1	2	3	4	5	
20						
30						
40						
50						
60						
70						
75						

4.5.4 Pengujian Variasi Jumlah Iterasi

Pada pengujian variasi jumlah iterasi ini bertujuan untuk mengetahui jumlah iterasi dari sistem terbaik yang sebaiknya digunakan oleh sistem. Berikut Tabel 4.22 akan menunjukkan mengenai tabel pengujian variasi jumlah iterasi.

Tabel 4.22 Pengujian Variasi Jumlah Iterasi

Jumlah Iterasi	Nilai MSE Percobaan ke-i					Rata - Rata MSE
	1	2	3	4	5	
20						
35						
50						
100						
150						
200						

BAB 5 IMPLEMENTASI

Pada Bab 5 yaitu bab implementasi akan dijelaskan secara rinci tentang proses implementasi berupa kode program dan antarmuka. Implementasi dilakukan berdasar pada hasil perancangan sistem yang telah dilakukan.

5.1 Implementasi Sistem

Pada Bab 4 telah dijelaskan mengenai hasil perancangan dan antarmuka yang akan di implementasikan oleh sistem. Selanjutnya pada Bab 5 ini, akan dijelaskan mengenai hasil implementasi sistem yang digunakan untuk memprediksi harga batubara. Implementasi sistem dilakukan dengan menggunakan aplikasi *netbeans* 8.1 dan menggunakan bahasa pemrograman *Java*. Pada Kode Program 5.1 akan ditunjukkan mengenai proses perubahan data ke dalam fitur.

5.1.1 Implementasi Proses Normalisasi Data

```

1. double[][] normalisasi(double data[][], double newmin,
2. double newmax) {
3.     double[] mins = getMin();
4.     double[] maxs = getMax();
5.     double data_normal[][] = new
6. double[data.length][data[0].length];
7.     for (int i = 0; i < data.length; i++) {
8.         for (int j = 0; j < data[0].length; j++) {
9.             data_normal[i][j] = ((data[i][j] -
10. mins[j]) / (maxs[j] - mins[j])) * (newmax
11. - newmin) + newmin;
12.         }
13.     }
14.     return data_normal;
15. }

```

Kode Program 5.1 Proses Normalisasi Data

Berikut merupakan penjelasan dari Kode Program 5.1 di atas:

1. Baris ke 1 hingga 2 merupakan tahap deklarasi method `double[][] normalisasi` berparameter `double data[]`, `double newmax`, `double newmin`.
2. Baris ke 3 hingga 4 merupakan pengambilan nilai minimal dengan `getMin` dan nilai maksimal dengan `getMax`.
3. Baris ke 5 hingga 6 deklarasi variabel `data_normal` yang mengambil nilai dari `data.length`.
4. Baris 7-11 merupakan proses perhitungan normalisasi.
5. Baris 14 proses *return* `data_normal`.

5.1.2 Implementasi Proses Inisialisasi Bobot dan Bias

```

1. double[][] inisialisasi_bobot_V(int jum_fitur, int
2. jum_hidden, double min, double max) {
3.     double range = max - min;
4.     double v[][] = new double[jum_fitur][jum_hidden];
5.     for (int i = 0; i < jum_fitur; i++) {
6.         for (int j = 0; j < jum_hidden; j++) {
7.             double weight = (Math.random() * range) +
8.             min;
9.             v[i][j] = weight;
10.        }
11.    }
12.    return v;
13. }

14.
15. double[] inisialisasi_bias_V_W(int jum_hidden, double
16. min, double max) {
17.     double range = (max - min);
18.     double bias_v[] = new double[jum_hidden + 1];
19.     for (int i = 0; i < jum_hidden; i++) {
20.         double temp = (Math.random() * range) + min;
21.         bias_v[i] = temp;
22.     }
23.     bias_v[bias_v.length - 1] = (Math.random() *
24.     (range)) + (min);
25.     return bias_v;
26. }

27.
28. double[] inisialisasi_bobot_W(int jum_hidden, double max,
29. double min) {
30.     double range = (max - min);
31.     double w[] = new double[jum_hidden];
32.     for (int i = 0; i < jum_hidden; i++) {
33.         double temp = (Math.random() * range) + min;
34.         w[i] = temp;
35.     }
36.     return w;
37. }

38. void inisialisasi_bobot_bias(int jum_fitur, int
39. jum_hidden, double min, double max) {
40.     this.v = inisialisasi_bobot_V(jum_fitur, jum_hidden,
41.     min, max);
42.     double[] temp_bias =
43.     inisialisasi_bias_V_W(jum_hidden, min, max);
44.     this.w = inisialisasi_bobot_W(jum_hidden, max, min);
45.     this.bias_w = temp_bias[temp_bias.length - 1];
46.     this.bias_v = new double[temp_bias.length - 1];
47. }

```

48	for (int i = 0; i < bias_v.length; i++) {
49	this.bias_v[i] = temp_bias[i];
50	}
51	}
52	

Kode Program 5.2 Proses Inisialisasi Bobot dan Bias

Berikut merupakan penjelasan dari Kode Program 5.2 proses inisialisasi bobot dan bias:

1. Baris ke 1 hingga 2, yaitu deklarasi method `double[][] inisialisasi_bobot_V` dengan parameter `int jum_fitur`, `int jum_hidden`, `double min` dan `double max`.
2. Baris ke 3 deklarasi variable `double range` untuk membuat range nilai bobot dari parameter `max` dan `min`.
3. Baris ke 4 deklarasi variabel `v` bertipe array 2 dimensi yang berisi parameter `jum_fitur` dan `jum_hidden`.
4. Baris ke 6 hingga 10 proses mendapatkan nilai bobot `v` secara random dengan range nilai -0.5 hingga 0.5
5. Baris ke 13 proses *return* nilai `v`.
6. Baris ke 16 hingga 17 deklarasi method `double[] inisialisasi_bobot_V_W` dengan parameter `int jum_hidden`, `double min` dan `double max`.
7. Baris ke 18 deklarasi variabel `range`.
8. Baris ke 19 deklarasi variabel `bias_v`.
9. Baris ke 20 hingga 25 proses mendapatkan nilai bobot bias `v` secara random dengan range nilai -0.5 hingga 0.5.
10. Baris ke 27 proses *return* `bias_v`.
11. Baris ke 30 deklarasi method `inisialisasi_bobot_W[]` dengan parameter `int jum_hidden`, `double min` dan `double max`.
12. Baris ke 32 deklarasi variabel `range`.
13. Baris ke 33 deklarasi variabel array `w`.
14. Baris ke 34 hingga 37 proses mendapatkan nilai `w` secara random dengan range nilai -0.5 hingga 0.5.
15. Baris ke 39 proses *return* nilai `w`.
16. Baris ke 41 hingga ke 42 merupakan deklarasi method `void inisialisasi_bobot_bias` yang berguna untuk memanggil method-method

sebelumnya, dengan variabel `int jum_fitur`, `int jum_hidden`, `double min`, dan `double max`.

17. Baris ke 43 merupakan inisialisasi variabel `this.v` yang berisi sama dengan method `inisialisasi_bobot_v`
18. Baris ke 45 hingga ke 46 deklarasi variabel array `temp_bias` untuk menyimpan nilai pada fungsi `inisialisasi_bias_V_W`.
19. Baris ke 47 merupakan inisialisasi variabel `this.w` yang berisi sama dengan method `inisialisasi_bobot_w`.
20. Baris ke 49 merupakan inisialisasi variabel `this.bias_w`.
21. Baris ke 50 merupakan inisialisasi variabel `this.bias_v`.

5.1.3 Implementasi Proses Feedforward

```

1. double feedforward(double[] data, double[][] v,
2. double[] bias, double[] w, double bias1) {
3.     double[] z_in = new double[v[0].length];
4.     double[] z = new double[z_in.length];
5.     double tot;
6.     //hitung z
7.     for (int j = 0; j < v[0].length; j++) {
8.         tot = 0;
9.         for (int k = 0; k < data.length; k++) {
10.            tot += data[k] * v[k][j];
11.        }
12.        z_in[j] = tot + bias[j];
13.        //aktivasi
14.        z[j] = 1 / (1 + Math.exp(-z_in[j]));
15.    }
16.    this.z = z;
17.
18.    //hitung y
19.    tot = 0;
20.    for (int i = 0; i < z.length; i++) {
21.        tot += z[i] * w[i];
22.    }
23.    tot += bias1;
24.    double y = 1 / (1 + Math.exp(-tot));
25.
26.    return y;
27. }
28.

```

Kode Program 5.3 Proses Feedforward

Berikut merupakan penjelasan dari Kode Program 5.3 proses Feedforward:

1. Baris ke 1 hingga 2 merupakan proses deklarasi method double feedforward dengan parameter `double[] data`, `double [][] v`, `double[] bias`, `double[] w` dan `double bias 1`.
2. Baris ke 3 proses deklarasi variabel `z_in` dengan mengambil nilai dari `v`.
3. Baris ke 4 deklarasi variabel `z` mengambil nilai dari `z_in`
4. Baris ke 5 deklarasi variabel `tot`.
5. Baris ke 7 hingga 12 merupakan proses perhitungan nilai `z_in`
6. Baris ke 14 merupakan proses perhitungan nilai `z[j]` menggunakan fungsi aktivasi.
7. Baris ke 20 hingga 23 untuk menghitung nilai `y`.
8. Baris ke 24 merupakan proses perhitungan nilai `y` dengan fungsi aktivasi.
9. Baris ke 26 proses pengembalian nilai `y`.



5.1.4 Implementasi Proses Backpropagation

```

1. void backward(double data_y, double t, double alfa,
2. double[] z, double data[], int jum_fitur) {
3.     double  $\delta$ [] = new double[z.length];
4.     double[] delta_w = new double[z.length];
5.
6.     //hitung  $\delta_k$ 
7.     double  $\delta_k$  = (t - data_y) * data_y * (1 -
8.     data_y);
9.     //hitung delta w
10.    for (int i = 0; i < delta_w.length; i++) {
11.        delta_w[i] =  $\delta_k$  * alfa * z[i];
12.    }
13.    this.delta_w = delta_w;
14.
15.    //hitung delta bias w
16.    double delta_bias_w =  $\delta_k$  * alfa * this.bias_w;
17.    this.delta_bias_w = delta_bias_w;
18.
19.    //hitung  $\delta$ 
20.    double  $\delta_{in}$ [] = new double[z.length];
21.    for (int i = 0; i <  $\delta_{in}$ .length; i++) {
22.         $\delta_{in}[i]$  =  $\delta_k$  * this.w[i];
23.    }
24.    for (int i = 0; i < z.length; i++) {
25.         $\delta[i]$  =  $\delta_{in}[i]$  * z[i] * (1 - z[i]);
26.    }
27.    //hitung delta v
28.    double    delta_v[][] = new
29. double[jum_fitur][z.length];
30.    for (int i = 0; i < jum_fitur; i++) {
31.        for (int j = 0; j < z.length; j++) {
32.            delta_v[i][j] = alfa *  $\delta[j]$  * data[i];
33.        }
34.    }
35.    this.delta_v = delta_v;
36.    //hitung delta bias v
37.    double[] delta_bias_v = new double[z.length];
38.    for (int i = 0; i < delta_bias_v.length; i++)
39.    {
40.        delta_bias_v[i] = alfa *  $\delta[i]$ ;
41.    }
    this.delta_bias_v = delta_bias_v;

```

Kode Program 5.4 Proses Backpropagation

Berikut adalah penjelasan dari Kode Program 5.4 proses Backpropagation:

1. Baris ke 1 hingga 2 merupakan deklarasi method void backward dengan parameter double data_y, double t, double alfa, double[] z, double data[], dan int jum_fitur.
2. Baris ke 3 merupakan proses deklarasi variabel array δ dengan mengambil nilai dari z.length.
3. Baris ke 4 proses deklarasi variabel delta_w.
4. Baris ke 7 hingga 8 merupakan proses untuk perhitungan nilai δ_k .
5. Baris ke 10 hingga ke 3 merupakan proses untuk perhitungan nilai delta_w.
6. Baris ke 16 hingga 17 adalah proses untuk perhitungan delta_bias_w.
7. Baris ke 20 hingga ke 22 adalah proses untuk perhitungan nilai δ_{in} .
8. Baris ke 24 hingga ke 25 merupakan proses untuk perhitungan nilai δ .
9. Baris ke 28 hingga ke 29 merupakan deklarasi variabel array delta_v.
10. Baris ke 30 hingga ke 35 merupakan proses untuk perhitungan delta_v.
11. Baris ke 37 merupakan deklarasi variabel array delta_bias_v.
12. Baris ke 38 hingga ke 42 merupakan proses untuk perhitungan delta_bias_v.

5.1.5 Implementasi Proses Update Bobot dan Bias

```

1.  double[] update_bias_v(double bias_v[]) {
2.      double new_bias_v[] = new double[bias_v.length];
3.      for (int i = 0; i < new_bias_v.length; i++) {
4.          new_bias_v[i] = bias_v[i] + this.delta_bias_v[i];
5.      }
6.      return new_bias_v;
7.  }
8.
9.      double update_bias_w(double bias_w) {
10.         return bias_w + this.delta_bias_w;
11.     }
12.
13.     double[][] update_v(double v[][]) {
14.         double[][] new_v = new double[v.length][v[0].length];
15.         for (int i = 0; i < new_v.length; i++) {
16.             for (int j = 0; j < new_v[0].length; j++)

```

Kode Program 5.5 Proses Update Bobot dan Bias

Berikut merupakan penjelasan dari Kode Program 5.5 proses update bobot dan bias:

1. Baris ke 1 merupakan deklarasi method double[] update_bias_v dengan parameter double bias_v[].

```

20. {
21.     new_v[i][j] = v[i][j] +
22.     this.delta_v[i][j];
23. }
24. }
25.     return new_v;
26. }
27.
28. double[] update_w(double w[]) {
29.     double new_w[] = new double[w.length];
30.     for (int i = 0; i < new_w.length; i++) {
31.         new_w[i] = w[i] + this.delta_w[i];
32.     }
33.     return new_w;
34. }

```

2. Baris ke 2 merupakan deklarasi variabel array `new_bias_v` mengambil nilai dari `bias_v.length`.
3. Baris ke 4 hingga ke 6 merupakan proses untuk perhitungan `new_bias_v`.
4. Baris ke 8 merupakan pengembalian nilai `new_bias_v`.
5. Baris ke 11 merupakan deklarasi method `double update_bias_w` dengan parameter `bias_w`.
6. Baris ke 12 merupakan proses pengembalian nilai `bias_w` sekaligus menambahkan variabel `delta_bias_w` yang digunakan untuk proses perhitungan pada method `update_bias_w`.
7. Baris ke 15 merupakan deklarasi method `double[][] update_v` berisi parameter `double v[][]`.
8. Baris ke 16 merupakan deklarasi variabel array `double[][] new_v` berisi `v.length` dan `v[0].length`.
9. Baris ke 17 hingga ke 22 merupakan proses untuk perhitungan `new_v`.
10. Baris ke 25 merupakan proses pengembalian nilai `new_v`.
11. Baris ke 28 merupakan deklarasi method `double[] update_w` berparameter `double w[]`.
12. Baris ke 29 merupakan deklarasi variabel array `double[] new_w` berisi `w.length`.
13. Baris ke 30 hingga ke 31 merupakan proses untuk perhitungan `new_w`.
14. Baris ke 33 merupakan proses pengembalian nilai `new_w`.

5.1.6 Implementasi Proses *Training*

```

1. void training(double data_latih[][], double alfa,
2. double t[], int jum_fitur, int jum_hidden, double
3. max_error, double min, double max, int max_iter) {
4.     int index = 0;
5.     mse_per_i = new double[max_iter];
6.     for (int i = 0; i < max_iter; i++) {
7.         mse_per_i[i] = 0;
8.     }
9.     do {
10.        System.out.println("iterasi ke-" + index);
11.        System.out.println("bobot v");
12.        for (int i = 0; i < this.v.length; i++) {
13.
14.        System.out.println(Arrays.toString(this.v[i]));
15.        }
16.        System.out.println("bias v");
17.
18.        System.out.println(Arrays.toString(this.bias_v));
19.        System.out.println("bobot w");
20.
21.        System.out.println(Arrays.toString(this.w));
22.        System.out.println("bias w");
23.        System.out.println(this.bias_w);
24.        double y[] = new
25.        double[data_latih.length];
26.        for (int i = 0; i < data_latih.length; i++)
27.        {
28.            //feed forward
29.            y[i] = feedforward(data_latih[i], this.v,
30.            this.bias_v, this.w, this.bias_w);
31.
32.        backward(y[i], t[i], alfa, this.z, data_latih[i],
33.        jum_fitur);
34.        if (i != data_latih.length - 1) {
35.            this.v = update_v(this.v);
36.            this.w = update_w(this.w);
37.            this.bias_v = update_bias_v(this.bias_v);
38.            this.bias_w = update_bias_w(this.bias_w);
39.            System.out.println("bobot baru");
40.            System.out.println("bobot v");
41.            for (int j = 0; j < this.v.length;
42.            j++) {

```

43.	<code>System.out.println(Arrays.toString(this.v[j]));</code>
44.	<code>}</code>
45.	<code>System.out.println("bias v");</code>
46.	<code>System.out.println(Arrays.toString(this.bias_v));</code>
47.	<code>System.out.println("bobot w");</code>
48.	

Kode Program 5.6 Proses *Training*

Pada method void *Training* ini berisi mengenai parameter-parameter yang digunakan pada proses *training* dan berisi fungsi untuk memanggil method-method yang telah dijabarkan pada proses *feedforward* dan *backpropagation*. Berikut merupakan penjelasan dari Kode Program 5.6 proses *Training*:

1. Baris ke 1 hingga ke 3 deklarasi method void training berparameter double data_latih[[]], double alfa, double t[], int jum_fitur, int jum_hidden, double max_eror, double min, double max, dan int max_iter.
2. Baris ke 6 hingga 7 merupakan proses untuk memperoleh nilai mse_per_i.
3. Baris ke 10 untuk menampilkan hasil iterasi ke sesuai index.
4. Baris ke 11 hingga ke 14 proses untuk menampilkan bobot v dalam bentuk string.
5. Baris ke 16 hingga ke 18 adalah proses untuk menampilkan bias v dalam bentuk string.
6. Baris ke 19 hingga ke 21 adalah proses untuk menampilkan bobot w dalam bentuk string.
7. Baris ke 22 hingga ke 23 adalah proses untuk menampilkan bias w.
8. Baris ke 24 merupakan deklarasi variabel array y yang berisi data_latih.length.
9. Baris ke 35 inialisasi variabel this.v sama dengan update_v
10. Baris ke 36 adalah proses untuk inialisasi variabel this.w sama dengan update_w.
11. Baris ke 37 adalah proses untuk inialisasi variabel this.bias_v sama dengan update_bias_v.
12. Baris ke 39 adalah proses untuk inialisasi variabel this.bias_w sama dengan update_bias_w.
13. Baris ke 46 adalah proses untuk menampilkan bobot v yang baru dalam bentuk string.
14. Baris ke 51 adalah proses untuk menampilkan bias v dalam bentuk string.
15. Baris ke 54 adalah proses untuk menampilkan bobot w dalam bentuk string.
16. Baris ke 55 hingga ke 56 adalah proses untuk menampilkan bias w dalam bentuk string.
17. Baris ke 59 inialisasi variabel mse yang berisi nilai t dan y.
18. Baris ke 60 adalah proses untuk menampilkan mse.
19. Baris ke 61 hingga ke 63 adalah proses if else mse>max_error.

5.1.7 Implementasi Proses Testing

```

1. double[] testing(double[][] data_uji) {
2.     double y[] = new double[data_uji.length];
3.     double[][] v = this.v;
4.     double[] w = this.w;
5.     double[] bias_v = this.bias_v;
6.     double bias_w = this.bias_w;
7.     for (int i = 0; i < data_uji.length; i++) {
8.         double temp_y = feedforward(data_uji[i], v,
9.         bias_v, w, bias_w);
10.        System.out.println(temp_y);
11.        y[i] = temp_y;
12.    }
13.    return y;
14. }

```

Kode Program 5.7 Proses Testing

Berikut merupakan penjelasan untuk kode program 5.7 proses *Testing*:

1. Baris ke 1 adalah deklarasi method `double[] testing` yang berisi parameter `double[][] data_uji`.
2. Baris ke 2 merupakan deklarasi variabel array `y` yang mengambil nilai dari `data_uji`.
3. Baris ke 3 adalah proses deklarasi variabel array `v` sama dengan `this.v`.
4. Baris ke 4 adalah proses deklarasi variabel `w` sama dengan `this.w`.
5. Baris ke 5 adalah proses deklarasi variabel array `bias_v` sama dengan `this.bias_v`.
6. Baris ke 7 adalah proses perulangan pada `data_uji.length`.
7. Baris ke 8 adalah proses inisialisasi variabel `temp_y` untuk menyimpan nilai yang berisi method `feedforward` dengan parameter `data_uji`, `v`, `bias_v`, `w`, `bias_w`.
8. Baris ke 10 hingga ke 11 adalah proses untuk menampilkan nilai `temp_y`.
9. Baris ke 13 merupakan proses pengembalian nilai pada `y`.

5.1.8 Implementasi Proses Denormalisasi Data

```

1. double[] denormalisasi(double[] data, double newmin,
2. double newmax) {
3.     double[] maxs = this.getMax();
4.     double[] mins = this.getMin();
5.     double data_denormal[] = new
6. double[data.length];
7.     for (int i = 0; i < data.length; i++) {
8.         data_denormal[i] = (int) Math.round(((data[i]
9. - newmin) / (newmax - newmin)) * (maxs[maxs.length -
10. 1] - mins[mins.length - 1]) + mins[mins.length - 1]);
11.     }
12.     return data_denormal;
13. }
14.

```

Kode Program 5.8 Proses Denormalisasi Data

Berikut merupakan penjelasan untuk kode program 5.8 proses denormalisasi data:

1. Baris ke 1 merupakan proses deklarasi method double[] denormalisasi dengan parameter double[] data, double newmin, double newmax.
2. Baris ke 3 merupakan proses inialisasi variabel array mins sama dengan this.getMax.
3. Baris ke 4 merupakan proses inialisasi variabel array mins sama dengan this.getMin.
4. Baris ke 5 merupakan proses deklarasi variabel array data_denormal[] mengambil nilai dari data.length.

5.1.9 Implementasi Evaluasi Sistem Menggunakan MSE

```

1. public double mse(double[] t, double[] y) {
2.     double tot = 0;
3.     for (int i = 0; i < t.length; i++) {
4.         double YminT = y[i] - t[i];
5.         tot += Math.pow(YminT, 2);
6.     }
7.     return tot / t.length;
8. }

```

Kode Program 5.9 Evaluasi Sistem Menggunakan MSE

Berikut merupakan penjelasan untuk kode program 5.9 proses evaluasi sistem menggunakan MSE:

1. Baris ke 1 merupakan proses deklarasi method double mse dengan parameter double[] t dan double[] y.

2. Baris ke 2 adalah proses inialisasi dari variabel tot bertipe data double.
3. Baris ke 3 hingga ke 5 adalah proses perhitungan untuk memperoleh nilai mse.
4. Baris ke 7 merupakan proses pengembalian nilai mse.

5.2 Implementasi Antarmuka

Pada bab 4 telah dilakukan proses perancangan antarmuka sistem Prediksi Harga Batubara Menggunakan Algoritma Backpropagation ini. Berdasarkan perancangan yang telah dilakukan, terdapat 7 bagian dalam 1 buah panel sistem.

Hasil pengujian yang ditampilkan adalah hasil prediksi, hasil bobot dan bias awal, MSE per iterasi, MSE akhir yang diperoleh dan hasil peramalan dari data actual. Gambar 5.1 , 5.2 serta 5.3 merupakan hasil antarmuka yang diperoleh dari sistem:

Input data:

System Prediksi Harga Batubara (Backpropagation)

Parameter Training dan Testing

Nilai Alpha: 0.5

Jumlah Neuron Hidden: 4

Jumlah Fitur: 3

Max Error: 0.001

Jumlah Iterasi: 50

Data Latih: 20

Pilih data: harga gunung bayan.xls [Pilih] [Proses]

Data Harga Batubara

Bulan	Gunung Bayan I (USD/ton)
September-2017	98.88
Agustus-2017	90.15
Juli-2017	84.72
Juni-2017	80.94
Mei-2017	89.98
April-2017	88.57

Bobot dan Bias Awal

Title 1 Title 2 Title 3 Title 4

MSE Per Iterasi

Title 1 Title 2 Title 3 Title 4

Hasil Peramalan - Data Aktual

Title 1 Title 2 Title 3 Title 4

[Prediksi]

Hasil Prediksi Bulan Selanjutnya

Gambar 5.1 Antarmuka Input Data

Hasil bobot dan bias awal serta MSE dari pelatihan:

System Prediksi Harga Batubara (Backpropagation)

Parameter Training dan Testing

Nilai Alpha: 0.5

Jumlah Neuron Hidden: 4

Jumlah Fitur: 3

Max Error: 0.001

Jumlah Iterasi: 50

Data Latih: 20

Pilih data:

Bobot dan Bias Awal

	0	1	2	3
Bobot v awal	0.30188275...	0.34987292...	-0.1141642...	0.2140357...
Bobot w awal	0.05967578...	0.27219601...	0.27046608...	0.2179450...
Bobot w awal	-0.4221445...	-0.0497490...	0.07308180...	-0.2452571...

MSE Per Iterasi

Iterasi ke-	MSE
1	0.06537776989023901
2	0.03259237563741964
3	0.02844623266765589
4	0.027893933277082374
5	0.02786756245035632
6	0.02788523412255598

MSE: 0.00211361848549167

Hasil Peramalan : Data Aktual

Data ke-	Data aktual (t)	Data Hasil (y)
1	61.0	69.0
2	62.0	69.0
3	63.0	70.0
4	63.0	70.0
5	64.0	70.0
6	65.0	70.0
7	69.0	70.0
8	73.0	70.0
9	67.0	71.0
10	68.0	70.0
11	69.0	70.0
12	70.0	70.0

Prediksi

Hasil Prediksi Bulan Selanjutnya

Data Harga Batubara

Bulan	Gunung Bayan I (USD/ton)
September-2017	98.88
Agustus-2017	90.15
Juli-2017	84.72
Juni-2017	80.94
Mei-2017	89.98
April-2017	88.57

Gambar 5.2 Antarmuka Training

Hasil prediksi:

System Prediksi Harga Batubara (Backpropagation)

Parameter Training dan Testing

Nilai Alpha: 0.5

Jumlah Neuron Hidden: 4

Jumlah Fitur: 3

Max Error: 0.001

Jumlah Iterasi: 50

Data Latih: 20

Pilih data:

Bobot dan Bias Awal

	0	1	2	3
Bobot v awal	0.30188275...	0.34987292...	-0.1141642...	0.2140357...
Bobot w awal	0.05967578...	0.27219601...	0.27046608...	0.2179450...
Bobot w awal	-0.4221445...	-0.0497490...	0.07308180...	-0.2452571...

MSE Per Iterasi

Iterasi ke-	MSE
1	0.06537776989023901
2	0.03259237563741964
3	0.02844623266765589
4	0.027893933277082374
5	0.02786756245035632
6	0.02788523412255598

MSE: 0.00211361848549167

Hasil Peramalan : Data Aktual

Data ke-	Data aktual (t)	Data Hasil (y)
1	61.0	69.0
2	62.0	69.0
3	63.0	70.0
4	63.0	70.0
5	64.0	70.0
6	65.0	70.0
7	69.0	70.0
8	73.0	70.0
9	67.0	71.0
10	68.0	70.0
11	69.0	70.0
12	70.0	70.0

Prediksi

Hasil Prediksi Bulan Selanjutnya

82.0

Data Harga Batubara

Bulan	Gunung Bayan I (USD/ton)
September-2017	98.88
Agustus-2017	90.15
Juli-2017	84.72
Juni-2017	80.94
Mei-2017	89.98
April-2017	88.57

Gambar 5.3 Antarmuka Hasil Prediksi

BAB 6 PENGUJIAN

Bab ini berisi mengenai proses pengujian dari sistem yang telah selesai dilakukan. Pengujian yang dilakukan adalah pengujian parameter backpropagation yang meliputi pengujian jumlah neuron *input* dan neuron *hidden*, jumlah *learning rate* (α), jumlah data *training* dan jumlah iterasi.

1.1 Pengujian Jumlah Neuron *Input* dan *Hidden* Terbaik.

Pengujian jumlah neuron *input* dan *hidden* ini bertujuan untuk memperoleh arsitektur *backpropagation* terbaik berdasarkan nilai *error* yang dihitung menggunakan MSE. Data yang digunakan pada pengujian ini adalah data harga batubara jenis Gunung Bayan I sejumlah 105 data. Neuron *input* atau dapat dikatakan sebagai jumlah fitur yang digunakan adalah 5,7,10 dan 15 sedangkan untuk jumlah neuron *hidden* yaitu dari 1 hingga 10. Data *training* yang digunakan untuk pengujian ini sebanyak 70 data dan 12 data *testing*. *Learning rate* yang digunakan dalam pengujian ini sebesar 0.1 dengan max error sebesar 0.001. Jumlah iterasi yang digunakan sebanyak 20 kali. Dilakukan 5 kali percobaan *running* program, kemudian akan dicari rata-rata nilai MSE nya.

Percobaan 1:

Tabel 6.1 Nilai *Error* Percobaan Pertama

Jumlah Neuron Hidden	Jumlah Neuron Input			
	5	7	10	15
1	0.090967	0.071007	0.02411772	0.013681
2	0.102267	0.0670897	0.01987356	0.0119001
3	0.096398	0.0648614	0.01347501	0.0141363
4	0.089821	0.0492027	0.01499988	0.0153803
5	0.077974	0.0661017	0.01569565	0.0150713
6	0.087663	0.0612868	0.0059473	0.0169826
7	0.097882	0.0532543	0.01055281	0.0142633
8	0.075781	0.0523019	0.00787518	0.0165637
9	0.065986	0.0576584	0.00491114	0.0181353
10	0.071652	0.0335688	0.00334603	0.0180556

Percobaan 2:

Tabel 6.2 Nilai Error Percobaan Kedua

Jumlah Neuron Hidden	Jumlah Neuron Input			
	5	7	10	15
1	0.0750776	0.07741991	0.021489207	0.01456576
2	0.1079268	0.07520554	0.024206242	0.01242982
3	0.097118	0.06615077	0.011555603	0.01356886
4	0.0930083	0.06436902	0.012929246	0.01276826
5	0.084741	0.05903368	0.005321927	0.01446897
6	0.0932826	0.05757764	0.016350866	0.01480648
7	0.0960433	0.05003058	0.006415493	0.01916402
8	0.0723254	0.04402998	0.005371936	0.0181317
9	0.0882505	0.04920231	0.006572468	0.01753888
10	0.0823141	0.03675315	0.005239636	0.02483348

Percobaan 3:

Tabel 6.3 Nilai Error Percobaan Ketiga

Jumlah Neuron Hidden	Jumlah Neuron Input			
	5	7	10	15
1	0.104474	0.05415404	0.016144217	0.01920123
2	0.0972355	0.07357813	0.020368679	0.01540632
3	0.1074028	0.07810823	0.011113359	0.01442715
4	0.0982343	0.06512287	0.028910388	0.01690384
5	0.0851338	0.05120886	0.015916461	0.01935487
6	0.0915302	0.05258177	0.01065521	0.0129241
7	0.0950226	0.06040746	0.006671824	0.02030972
8	0.0743866	0.04820797	0.003874402	0.0193739
9	0.0770815	0.03724706	0.003828918	0.01610363
10	0.0776068	0.03890863	0.003815591	0.01696642

Percobaan 4:

Tabel 6.4 Nilai Error Percobaan Keempat

Jumlah Neuron Hidden	Jumlah Neuron Input			
	5	7	10	15
1	0.1112648	0.0727082	0.020588293	0.01183423
2	0.10027	0.07131413	0.014726878	0.01217744
3	0.1026494	0.05896985	0.014984857	0.01054085
4	0.1000111	0.06885718	0.017660478	0.19887068

5	0.0856632	0.06452283	0.012377709	0.01690786
6	0.0894746	0.05562454	0.008609951	0.01404568
7	0.0856192	0.05184837	0.007186031	0.02236039
8	0.0711701	0.0537387	0.007168566	0.01609136
9	0.0830261	0.0548853	0.006421988	0.02250336
10	0.0593273	0.04119068	0.004056809	0.0161927

Percobaan 5:

Tabel 6.5 Nilai Error Percobaan Kelima

Jumlah Neuron Hidden	Jumlah Neuron Input			
	5	7	10	15
1	0.1047468	0.07733725	0.020520614	0.01353213
2	0.1084533	0.07172702	0.023170983	0.01307297
3	0.0997052	0.06938846	0.011012355	0.01641305
4	0.0951227	0.05976924	0.013927323	0.01228956
5	0.0802752	0.04895127	0.007104476	0.01387336
6	0.0936706	0.04211859	0.016090839	0.0155734
7	0.0809138	0.04804662	0.005900965	0.01754425
8	0.0820759	0.04620406	0.003296513	0.01836635
9	0.0746187	0.04380571	0.004715166	0.02355302
10	0.0564184	0.04692506	0.005802406	0.01974885

Rata-rata MSE:

Tabel 6.6 Rata-Rata Nilai Error

Jumlah Neuron Hidden	Jumlah Neuron Input			
	5	7	10	15
1	0.097306	0.07052528	0.020572011	0.01456288
2	0.1032305	0.07178291	0.020469268	0.01299733
3	0.1006547	0.05361805	0.012428236	0.01381724
4	0.0952394	0.0614642	0.017685464	0.05124253
5	0.0827575	0.05796366	0.011283245	0.01593526
6	0.0911241	0.05383787	0.011530834	0.01486646
7	0.0910962	0.05271746	0.007345424	0.01872834
8	0.0751479	0.04889653	0.005517319	0.01770541
9	0.0777925	0.04855976	0.005289936	0.01956684
10	0.0694638	0.03946926	0.004452094	0.01915942

Dari hasil rata-rata perhitungan nilai *error* menggunakan MSE dapat diperoleh hasil bahwa jumlah neuron *input* dan neuron hidden mempengaruhi nilai *error* yang dihasilkan oleh sistem. Nilai *error* terendah diperoleh oleh pasangan jumlah neuron

input sebesar 10 neuron dan jumlah neuron *hidden* 10 neuron. Kombinasi jumlah neuron *input* dan *hidden* ini akan digunakan pada proses pengujian selanjutnya yaitu pengujian *learning rate*.

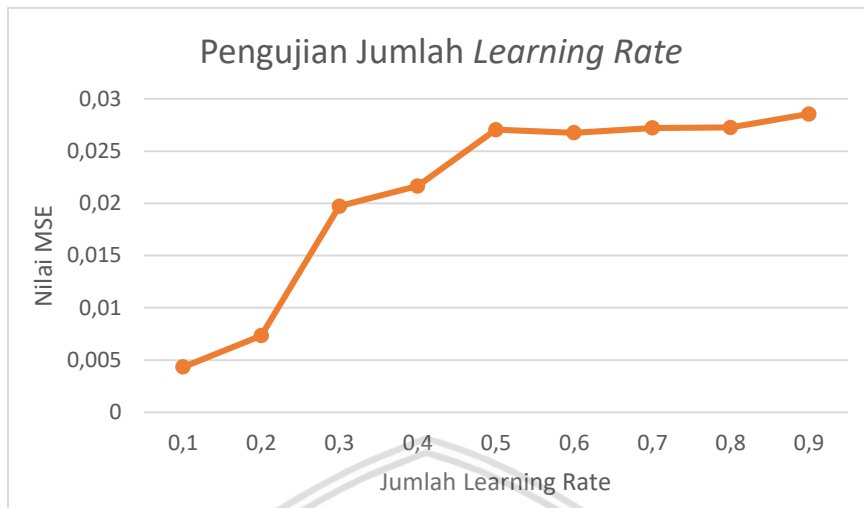
1.2 Pengujian *Learning Rate* Terbaik

Pada pengujian ini akan dicari nilai *learning rate* yang menghasilkan nilai error terendah. Pengujian akan dilakukan dengan menggunakan nilai *learning rate* dari 0,1 hingga 0,9. Jumlah neuron *input* sebesar 10 neuron dan neuron *hidden* sebesar 10 neuron yang diperoleh dari hasil pengujian neuron *input* dan neuron *hidden* terbaik. Pengujian ini akan dilakukan sebanyak 5 kali percobaan *running* program kemudian akan dicari rata-rata nilai *error* terendah. Data yang digunakan yaitu sebesar 70 data *training* dan 12 data *testing* dengan total data sebanyak 105 data. Hasil dari pengujian *learning rate* terbaik dapat dilihat pada Tabel 6.7.

Tabel 6.7 Hasil Pengujian Nilai *Learning Rate* Terbaik

Jumlah Learning Rate	Nilai MSE Percobaan ke-i					Rata -Rata MSE
	1	2	3	4	5	
0,1	0.0044567	0.003997	0.00393	0.004086	0.005256	0.00434511
0,2	0.0088312	0.005621	0.004379	0.010999	0.006836	0.00733345
0,3	0.0186999	0.01662	0.021601	0.018605	0.023117	0.01972846
0,4	0.022904	0.024593	0.021642	0.019096	0.020051	0.02165702
0,5	0.0230197	0.028371	0.029623	0.028632	0.025601	0.02704941
0,6	0.0305159	0.02695	0.028956	0.02807	0.019298	0.02675796
0,7	0.0209635	0.024643	0.027794	0.031012	0.031654	0.02721337
0,8	0.026591	0.03066	0.025554	0.025406	0.028097	0.02726149
0,9	0.0342054	0.026684	0.024234	0.031009	0.026629	0.02855236

Pada Tabel 6.7 dapat dilihat bahwa nilai *learning rate* mempengaruhi tingkat nilai error yang dihasilkan oleh sistem. Nilai *error* terendah yang dihasilkan sebesar 0.00434511 dimiliki oleh nilai *learning rate* sebesar 0,1. Semakin besar nilai *learning rate* nilai *error* yang dihasilkan semakin besar. Nilai *learning rate* terbaik akan digunakan pada pengujian selanjutnya yaitu pengujian jumlah data *training*. Berikut merupakan grafik dari pengujian *learning rate* pada Gambar 6.1.



Gambar 6.1 Grafik Hasil Pengujian Jumlah *Learning Rate*

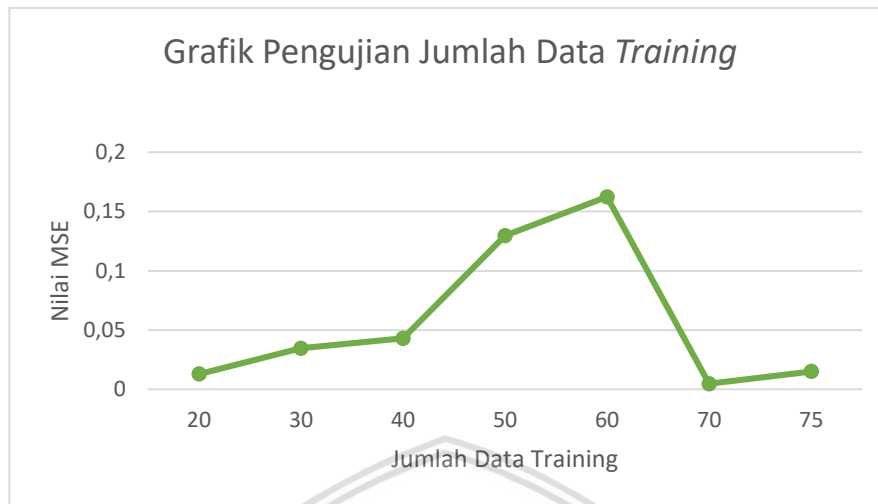
1.3 Pengujian Variasi Jumlah Data *Training*

Pengujian jumlah data latih (*training*) dilakukan untuk mengetahui apakah ada pengaruh nilai *error* terhadap variasi jumlah data *training* yang digunakan. Pada pengujian ini jumlah data *training* yang akan diuji adalah sebanyak 20, 30, 40, 50, 60, 70 dan 75 data *training*. Sedangkan untuk data *testing* yang digunakan jumlah nya tetap yaitu 12 data. Setiap kombinasi pola data *training* dan data *testing* akan dilakukan 5 kali percobaan *running* program. Tabel 6.8 berikut akan menunjukkan hasil dari pengujian ini:

Tabel 6.8 Hasil Pengujian Data *Training*

Jumlah Data Latih	Nilai MSE Percobaan ke-i					Rata -Rata MSE
	1	2	3	4	5	
20	0.0134997	0.013335	0.01286	0.012151	0.0122305	0.01281516
30	0.0361553	0.034122	0.035263	0.033287	0.0342701	0.03461934
40	0.0433939	0.041902	0.046458	0.042631	0.0406154	0.04300013
50	0.1199586	0.125805	0.124208	0.142786	0.1353025	0.12961203
60	0.1692607	0.154818	0.14692	0.186018	0.1543376	0.16227093
70	0.0036446	0.005621	0.003224	0.006091	0.0046996	0.00465612
75	0.0132751	0.01248	0.016616	0.016652	0.0161906	0.01504292

Grafik hasil pengujian pengaruh jumlah data *training* terhadap nilai *error* yang dihasilkan oleh sistem, ditunjukkan pada Gambar 6.2.



Gambar 6.2 Grafik Hasil Pengujian Jumlah Data Training

Dapat dilihat dari gambar grafik di atas bahwa variasi jumlah data *training* mempengaruhi hasil nilai *error*. Jumlah nilai *error* tertinggi dihasilkan oleh jumlah data *training* sebesar 60 data. Sedangkan untuk jumlah data *training* yang menghasilkan nilai *error* terendah yaitu sebanyak 70 data. Dari hasil grafik di atas dapat disimpulkan bahwa penggunaan lebih banyak data *training* bukan berarti nilai *error* yang dihasilkan semakin sedikit. Hal ini ditunjukkan pada jumlah data *training* sebanyak 75 data mengalami sedikit kenaikan nilai *error* yang dihasilkan.

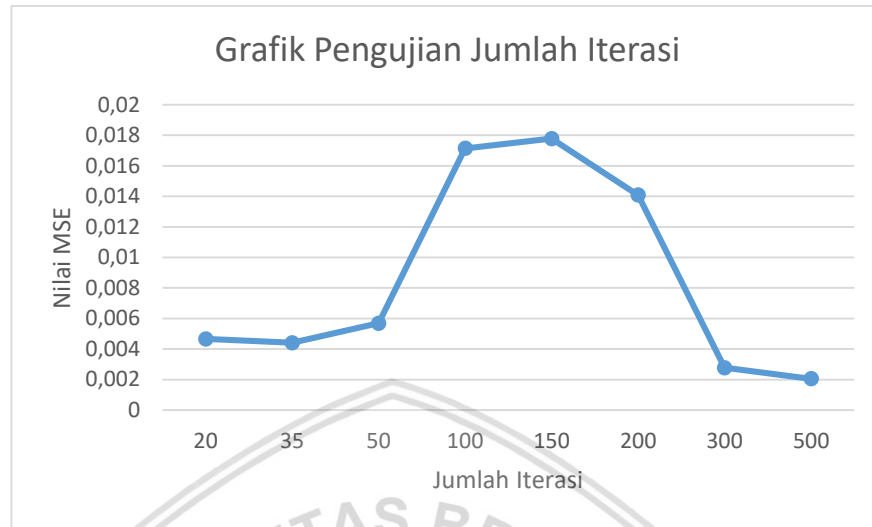
1.4 Pengujian Variasi Jumlah Iterasi

Pengujian variasi dari jumlah iterasi bertujuan untuk memperoleh jumlah iterasi terbaik pada arsitektur *backpropagation* yang digunakan. Pada pengujian ini menggunakan parameter terbaik dari pengujian-pengujian sebelumnya yaitu jumlah neuron *input* dan neuron *hidden* terbaik, jumlah *learning rate* terbaik dan jumlah data *training* terbaik. Hasil pengujian jumlah iterasi dapat dilihat pada Tabel 6.9 berikut.

Tabel 6.9 Hasil Pengujian Jumlah Iterasi

Jumlah Iterasi	Nilai MSE Percobaan ke-i					Rata -Rata MSE
	1	2	3	4	5	
20	0,0036446	0,005621	0,003224	0,006091	0,0047	0,00465612
35	0,0056021	0,003487	0,003491	0,003784	0,005688	0,00441058
50	0,0037194	0,004278	0,008817	0,005939	0,005673	0,00568542
100	0,0167864	0,018933	0,01736	0,015568	0,017077	0,01714473
150	0,0156779	0,016656	0,017764	0,018108	0,020708	0,01778293
200	0,011567	0,013457	0,014116	0,014577	0,016686	0,01408053
300	0,002624	0,00233	0,003564	0,002874	0,002498	0,00277804
500	0,0021702	0,002244	0,002046	0,00193	0,001874	0,00205284

Grafik hasil dari pengujian jumlah iterasi dapat dilihat pada Gambar 6.3 berikut:



Gambar 6.3 Grafik Hasil Pengujian Jumlah Iterasi

Dari grafik pada gambar di atas dapat dilihat bahwa jumlah iterasi mempengaruhi tingkat nilai *error* yang dihasilkan oleh sistem. Semakin banyak jumlah iterasi tidak berarti bahwa nilai *error* yang dihasilkan semakin rendah. Rata-rata nilai *error* terendah diperoleh oleh jumlah iterasi sebanyak 500 iterasi.

BAB 7 PENUTUP

7.1 Kesimpulan

Setelah dilakukan tahap implementasi dan pengujian dari penelitian Implementasi Metode *Backpropagation* untuk Prediksi Harga Batu Bara ini, maka dapat disimpulkan bahwa:

1. Dari hasil Implementasi sistem serta pengujian yang telah dilakukan pada penelitian diperoleh arsitektur jaringan *backpropagation* terbaik berdasarkan pengujian parameter *backpropagation* sebagai berikut:
 - a. 10 neuron pada *input layer* dan 10 neuron pada *hidden layer*.
 - b. Nilai *learning rate* terbaik yaitu 0,1.
 - c. Jumlah data *training* terbaik sebanyak 70 data.
 - d. Jumlah iterasi terbaik yaitu sebanyak 500 iterasi.
2. Rata-rata nilai *error* terendah yang mampu dihasilkan oleh sistem setelah dilakukan pengujian jumlah iterasi sebagai pengujian terakhir yaitu sebesar 0,00205284 sedangkan rata-rata nilai *error* tertinggi sebesar 0.01778293. Dapat disimpulkan bahwa sistem sudah cukup baik dalam melakukan prediksi berdasarkan nilai *error* yang dihasilkan.
3. Pada pengujian sistem ditemukan bahwa jumlah data *training* dengan nilai *error* terendah yaitu sebanyak 70 data, namun terjadi peningkatan nilai *error* jika digunakan 75 data dan 78 data sebagai nilai data *training* maksimal. Sedangkan untuk data *testing* atau data uji yang digunakan adalah sebanyak 12 data.

7.2 Saran

Terdapat beberapa saran yang dapat dipergunakan sebagai bahan pertimbangan guna penelitian yang selanjutnya, diantaranya:

1. Jumlah data *training* maksimal yang dapat digunakan sebanyak 78 data. Hal ini dikarenakan keterbatasan jumlah data yang tersedia. Apabila menggunakan data *training* yang lebih banyak maka sistem akan mampu mengenali pola jaringan lebih akurat.
2. Dapat ditambahkan metode optimasi pada pencarian bobot dan bias *backpropagation*, pada penelitian ini bobot dan bias di peroleh secara *random*.
3. Input yang digunakan yaitu data harga batu bara pada bulan-bulan sebelum bulan yang ingin diprediksi. *Output* yang dihasilkan yaitu prediksi harga 1 bulan kedepan. Apabila ingin mengetahui hasil dari 2 bulan selanjutnya maka

perlu dilakukan *update* data terlebih dahulu. *Output* yang dihasilkan dapat lebih beragam lagi. Misalnya, prediksi harga batu bara dalam setahun ke depan. Jadi ada 12 *output* data yang dihasilkan.



DAFTAR PUSTAKA

- Aritonang, R. Lerbin. 2002. Peramalan Bisnis Edisi 1. Jakarta : Penerbit Ghalia Indonesia.
- BP, 2017. *Statistical Review of World Energy*. [Online] Available at : <https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy.html> [Accessed 12 1 2018].
- Bumi, B. S. 2016. Implementasi Metode AI- Alaoui Backpropagation untuk Prediksi Harga Emas. *SKRIPSI Universitas Brawijaya*.
- BUMN, 2017. *Batubara di Indonesia (Produksi & Ekspor Batubara di Indonesia)*. [Online] Available at : <http://www.bumn.go.id/bukitasam/berita/2-Batubara-di-Indonesia-Produksi--Ekspor-Batubara-Indonesia-> [Accessed 12 1 2018].
- Chandra, M., Sovia, R. & Permana, R., 2015. *Analisis Metode Backpropagation Untuk Memprediksi Indeks Harga Saham Indofood Sukses Makmur Tbk. (INDF)*. Jurnal KomTekInfo, Volume 2(1).
- Chaturvedi, A. 2015. Rainfall Prediction using Back-Propagation Feed Forward Network. *International Journal of Computer Applications* (0975 – 8887) Volume 119 – No.4.
- Diessel, C.F.K., 1986. On the correlation between coal facies and depositional environments. 20th Newcastle Symposium on “Advances in the Study of the Sydney Basin”, Proceedings. Publication 246, Department of Geology, University of Newcastle, Australia.
- ESDM, 2017. *The 5th Indonesia-China Energy Forum*. [Online] Available at : <https://www.esdm.go.id/id/publikasi/the-5th-indonesia-china-energy-forum> [Accessed 15 1 2018].
- Fausset, L., 1994. *Fundamentals of Neural Network: Architecture, Algorithm, and Application*. New Jersey: Prentice Hall.
- Graupe, Daniel. 2009. *Principles of Artificial Neural Networks : 2nd*. World Scientific Publishing Co, Pte, Ltd.
- Graupe, Daniel. 2013. *Principles of Artificial Neural Networks : 3rd*. World Scientific Publishing Co, Pte, Ltd.
- Haykin, Simon. 1999. *Neural Networks A Comprehensive Foundation*. Prentice-Hall Inc., USA.
- Haykin, Simon. 2009. *Neural Networks and Learning Machine : Third Edition*. Pearson Education Inc., New Jersey.

- Heizer, J., Render, B. 1996. *Production and Operations Management : Strategic and Tactical Decisions*. Prentice-Hall Inc., USA.
- Heizer, J., Render, B. 2001. Prinsip-prinsip Manajemen Operasi : Operations Management. Jakarta : Salemba Empat.
- Indonesia-Investment. 2017. *Batubara*. [Online] Available at: <https://www.indonesia-investments.com/id/bisnis/komoditas/batubara/item236?> [Accessed 12 1 2018].
- Julpan, Nababan, E. B., Zarlis, M. 2015. Analisis Fungsi Aktivasi Sigmoid Biner dan Sigmoid Bipolar Dalam Algoritma Backpropagation pada Prediksi Kemampuan Siswa. *Jurnal Teknovasi*, Vol. 02 No. 1.
- Kholis, I., Alam, S. 2016. Analisis Variasi Parameter Backpropagation Artificial Neural Network dan Principal Component Analysis Terhadap Sistem Pengenalan Wajah. *Jurnal Teknik dan Ilmu Komputer Vol. 05 No.20*.
- Kriesel, D. 2007. A Brief Introduction to Neural Networks. Publisher: http://www.dkriesel.com/en/science/neural_networks
- Lahmiri, S. 2014. Wavelet Low and High Frequency Components as Features for Predicting Stock Prices with Backpropagation Neural Networks. *Journal of King Saud University – Computer and Information Sciences* (2014) 26, 218–227.
- Muchijidin. 2006. Pengendalian Mutu Dalam Industri Batu Bara. Bandung : ITB
- Nurmahaludin, 2014. Analisis Perbandingan Metode Jaringan Syaraf Tiruan Dan Regresi Linear Berganda Pada Prakiraan Cuaca. *Jurnal INTEKNA*, no. 2.
- Prijono, Achmad, dkk., 1992. Pengertian Batubara. [Online] Available at : <http://www.ptba.co.id/id/read/getting-to-know-coal>. [Accessed 12 1 2018].
- Permen-ESDM. 2017. *Paparan PerMen tahun 2017*. [Online] Available at: <http://jdih.esdm.go.id/peraturan/Permen%20ESDM%20Nomor%2007%20Tahun%202017.pdf>. [Accessed 13 1 2018].
- Sukandarrumidi. 2008. Batubara dan Gambut. Yogyakarta : Gadjah Mada University Press.
- Suryaningrum, D. A., Ratnawati, D. E., & Setiawan, B. D. 2017. Prediksi Waktu Panen Tebu Menggunakan Gabungan Metode *Backpropagation* dan Algoritma Genetika. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. I No. 11.
- Susanti, N. 2014. Penerapan Model Neural Network Backpropagation untuk Prediksi Harga Ayam. *Prosiding SNATIF ke-1*.
- Tanjung, D. H. 2015. Jaringan Syaraf Tiruan dengan Backpropagation untuk Memprediksi Penyakit Asma

- Triyono, A., Santoso, A. J., Pranowo. 2016. Penerapan Metode Jaringan Syaraf Tiruan Untuk Meramalkan Harga Saham (IHSG). *JURNAL SISTEM DAN INFORMATIKA* Vol. 11, No. 1.
- Wanto, A., Windarto, A.P., 2017. Analisis Prediksi Indeks Harga Konsumen Berdasarkan Kelompok Kesehatan Dengan Menggunakan Metode *Backpropagation*. *Jurnal dan Penelitian Teknik Informatika* Volume 2 Nomor 2.
- Yuniastari, N. L. A. K., Wirawan, I. G. P. 2014. Peramalan Permintaan Produk Perak Menggunakan Simple Moving Average dan Exponential Smoothing. *Jurnal Sistem dan Informatika* Vol.9 No. 1.

